

Tartu Ülikool

Loodus- ja täppisteaduste valdkond

Tehnoloogiainstituut

Mattias Tõnisson

**Veebilehe ligipääsetavus Eesti Puuetega Inimeste Koja veebilehe
arendusprotsessi näitel**

Bakalaureusetöö (12 EAP)

Arvutitehnika eriala

Juhendaja:

emeritiidotsent Helle Hein

Tartu 2020

Resümee/Abstract

Veebilehe ligipääsetavus Eesti Puuetega Inimeste Koja veebilehe arendusprotsessi näitel

Bakalaureusetöö eesmärk on tutvustada veebi ligipääsetavuse parendamise võimalusi veebikeskkonnas Eesti Puuetega Inimeste Koja (EPIKoda) veebilehe uuendamise praktilisel näitel. Esimeses peatükis arutletakse erivajadustega internetitarbijate kasutajakogemust parandavate tehnikate ning nende senise rakendamise edukuse üle. Järgnevalt analüüsitakse põgusalt veebiarendajate jaoks ligipääsetava sisu loomisel jätkuvalt aktuaalsete probleemide tagamaid. Töö viimases peatükis hinnatakse valminud veebilehe ligipääsetavust, jõudlust ja turvalisust ning kirjeldatakse edasisi arendusvõimalusi EPIKojaga jätkuvas koostöös.

CERCS: T120 Süsteemitehnoloogia, arvutitehnoloogia

Märksõnad: veebi ligipääsetavus, veebiarendus, erivajadused, WCAG, Kirby CMS

Web accessibility on the basis of developing a website for the Estonian Chamber of Disabled People

The aim of this thesis is to introduce some of the nuances of web accessibility based on the modernization of the website of EPIKoda (*Estonian Chamber of Disabled People*). The first chapter contains discussion about the different techniques which improve the user experience of people with special needs and how successfully they have been historically implemented. Following that, some of the reasons behind the continuously prevalent problem of inaccessible web being developed will be briefly analysed. In the last chapter of the thesis, the finished website will be benchmarked on accessibility, performance and security, and future plans concerning the continuing collaboration with EPIKoda will be described.

CERCS: T120 Systems engineering, computer technology

Keywords: web accessibility, web development, special needs, WCAG, Kirby CMS

Olulisemad lühendid, mõisted

UI (ingl *User Interface*) – kasutajaliides, osa veebilehest/-rakendusest, millega suhtleb kasutaja.

UX (ingl *User Experience*) – antud kontekstis kasutajakogemus veebilehe/-rakenduse kasutamisel, peamiselt kasutamise intuiitiivsus ja mugavus.

W3C (ingl *World Wide Web Consortium*) – rahvusvaheline organisatsioon, mille eesmärgiks on teha veeb täielikult erivajaduste-sõbralikuks.

WAI (ingl *Web Accessibility Initiative*) – W3C kollektiivne pingutus erivajadustega inimeste kasutajakogemuse parandamiseks.

WCAG (ingl *Web Content Accessibility Guidelines*) – veebimaterjalide ligipääsetavuse eeskirjad, WAI raames välja antav ligipääsetavuse tasemeid määrav abistav direktiiv. Määrab kolm nõuetele vastavuse taset: A (kõige madalam ja üldisem), AA ja AAA (kõrgemad ja spetsiifilisemad). Mida kõrgem on vastavuse tase, seda raskem on antud nõuet tehniliselt realiseerida ning seda rohkem mõjutab see lehekülje disainiprotsessi.

Ülemineku suhe (ingl *Conversion rate*) – suhe veebilehel pakutava teenuse/toote ostnud kasutajate ja lehele tulnud kasutajate vahel.

Nn “leivapuru rada” (ingl *Breadcrumb trail*) ehk jäljerida – horisontaalne menüü veebilehel, mis näitab kasutaja teekonda hetkel lahti oleva alaleheni ning võimaldab mugavalt eelnevatele lehtedele naasta.

Skaleeruv / dünaamiline veebidisain (ingl *Responsive web design*) – veebilehe kujunduse kohanemine vastavalt seadmete, akende või ekraanide suurustele ja/või muudele näitajatele.

Veebilehel olev funktsionaalsus (ingl *Functionality*) – kasutaja(te) toimingute tagajärjel tekkinud muudatused ja algatatud protsessid.

Struktuuripäringukeel (ingl SQL – *Structured Query Language*) – andmebaaside haldamiseks ning sealt päringute teel andmete kätte saamiseks kasutatav keel.

HTML (ingl *HyperText Markup Language*) – keel, mille abil ehitatakse üles veebirakenduste ja -lehtede sisu.

JS (ingl *Javascript*) – programmeerimiskeel, mida kasutatakse veebilehtede ja -rakenduste arenduses peamiselt selleks, et keskkond interaktiivseks ja dünaamiliseks teha.

PHP (ingl rekursiivne akronüüm *PHP Hypertext Preprocessor*) – HTML-ga tihedalt seotud skriptimiskeel, mis on mõeldud dünaamiliselt genereeritud veebilehtede kirjutamiseks. Peale paari spetsiifilise iseärasuse, on PHP süntaktiliselt väga sarnane C, Java ja Perl programmeerimiskeeltega.

Rakenduse programmeerimise kasutajaliides (ingl *API – Application Programming Interface*) - kujutab endast reeglitest koosnevat rakendusliidest, mille kaudu on kasutajal võimalik rakendusega suhelda.

Universaalne disain (ingl *universal design / UD*) on antud kontekstis veebikeskkonna kujundamine võimalikult kaasavalt, nii et seda saaksid kasutada võimalikult paljud inimesed nii suurel määral kui võimalik, ilma vajaduseta erikohtlemise või veebilehe ümber kohandamise jaoks (Koolmeister, 2016).

Ligipääsetav disain (ingl *accessible design*) on suunatud spetsiifiliselt puuetega inimestele või erivajadustega inimeste gruppidele. Kuna mõne kasutaja jaoks võib ligipääsetav disain olla ainus lahendus mingile konkreetsele probleemile, leidub situatsioone, kus selle kasutamine on vältimatu (Story, 2002).

Sisukord

Resümee/Abstract.....	2
Olulisemad lühendid, mõisted.....	3
Jooniste loetelu.....	7
1 Sissejuhatus	8
2 Ligipääsetavus.....	9
2.1 Ligipääsetavuse olemus ja tähtsus.....	9
2.2 Ligipääsetavuse ajaloost.....	10
3. Ligipääsetavuse parendamise tehnilised ja regulatoorsed aspektid	11
3.1 Ekraanilugered	11
3.2 Veebisisu optimeerimine.....	13
3.3 Probleemid ligipääsetavusega	15
3.3.1 WCAG 2.0 vs WCAG 2.1	15
3.3.2 Arendajate probleemide tagamaad	16
4 Metoodika.....	18
4.1 Tööprotsess.....	18
4.2 Kasutatud keeled	19
4.2.1 Struktuur ja funktsionaalsus	20
4.2.2 Kujundus	22
4.3 Kasutatud pistikprogrammid, teegid	24
4.3.1 jQuery.....	24
4.3.2 Matomo Web Analytics	25
4.3.3 Swiper-js.....	26
4.3.4 Küpsised ja js-cookie	28
4.3.5 Nested Accordion	29
4.4 Tähtsamad täidetud WCAG nõuded	29

4.4.1 Success Criterion 1.1.1 Non-text Content.....	30
4.4.2 Success Criterion 1.3.5 Identify Input Purpose.....	30
4.4.3 Success Criterion 1.4.3 Contrast	30
4.4.4 Success Criterion 1.4.12 Text Spacing.....	32
4.4.5 Success Criterion 2.1.3 Keyboard (No Exception)	32
4.5 Sisuhaldustarkvarad	34
4.5.1 WordPress	35
4.5.2 Kirby 3.....	35
5 Tulemused	38
5.1 Teoreetiline lahendus ligipääsetavuse probleemidele	38
5.2 EPIKoja kasutajate analüütika	40
5.3 Ligipääsetavuse, jõudluse ja turvalisuse hindamine	43
5.3.1 Automatiseeritud testid	43
5.3.2 Kasutajate tagasiside	44
6 Kokkuvõte	46
Kasutatud allikad.....	47
Litsents	53

Jooniste loetelu

Joonis 3.1: Optimaalne veebilehe struktuur HTML-elementidega (W3Schools 2020)	12
Joonis 3.2: Globaalselt kasutatavad lauaarvutite lahutusvõimed (StatCounter 2019)	14
Joonis 4.1: Veebilehitseja loodud DOM (Maldonado, 2018)	20
Joonis 4.2: Kirby CMS kujundusmalli kood	21
Joonis 4.3: EPIKoja veebilehe otsingumootori kood	22
Joonis 4.4: CSS kujundusreeglid <i>media-query</i> näitel	23
Joonis 4.5: jQuery teegi funktsionaalsus	25
Joonis 4.6: EPIKoja avalehel olev karussell	26
Joonis 4.7: Swiper-js teegi abil karusselli initsialiseerimine	27
Joonis 4.8: js-cookie teegi abil küpsise kasutamine	28
Joonis 4.9: Teksti liigendamine EPIKoja veebilehel Nested Accordion teegi abil	29
Joonis 4.10: Näide sobivast ja mittesobivast värvikontrastist (WebAIM, 2020)	31
Joonis 4.11: Korrektselt vormistatud teksti dünaamika (Kirkpatrick, 2018)	32
Joonis 4.12: EPIKoja veebilehe failisüsteem	36
Joonis 4.13: EPIKoja veebilehe halduspaneel	37
Joonis 5.1: EPIKoja külastajate arv 2019-2020	40
Joonis 5.2: EPIKoja külastamisel enim kasutatud veebilehitsejad	41
Joonis 5.3: EPIKoja külastamisel enim kasutatud operatsioonisüsteemid	42
Joonis 5.4: EPIKoja külastamisel enim kasutatud eraldusvõimed	42

1 Sissejuhatus

Eesti Puuetega Inimeste Koda on katusorganisatsioon, mis esindab kokku 285 erinevat erivajadustega inimestele suunatud organisatsiooni Eestis. EPIKoda kujutab endast erivajadustega inimeste valdkonnas üleriigilist koostööorganit ning suhtleb ja teeb koostööd ka paljude Euroopa tasandil tegutsevate katusorganisatsioonidega, sh Euroopa Puuetega Inimeste Foorum.

Kuna tegemist on erivajadustega inimestele suunatud organisatsiooniga, peab nende veebileht seda ka peegeldama, st olema ligipääsetav ja kasutatav kõigi poolt, keda nad esindavad. EPIKoja eelmine veebileht oli välimuselt vananenud, kannatas info ülekülluse all ning ei olnud skaleeruv erinevate seadmete jaoks. Sellepärast oli uue veebilehe arendamisel vaja tavalisest rohkem pöörata tähelepanu veebilehe ligipääsetavusele.

2019. aasta jaanuaris alustas töö autor koostöös veebidisaineri ning Tartu Ülikooli ajakirjanduse ja kommunikatsiooni õppekava üliõpilase Allan Augiga EPIKoja veebilehe uuendamisega. Eelmine veebileht oli ehitatud WordPress'i (WordPress Foundation, 2020) sisuhaldustarkvara peale, kust oli vaja kätte saada lehekülje sisu. Seejärel kujundati üle tõstetud sisu kasutades uus veebileht, seekord kasutades Kirby (Bastian Allgeier GmbH, 2020) sisuhaldustarkvara. Uus veebileht (ligipääsetav aadressil www.epikoda.ee) valmis 2019. aasta septembris.

Käesolev bakalaureusetöö koosneb neljast peatükist. Esimeses peatükis tutvustatakse veebi ligipääsetavuse temaatikat ja põgusalt ka selle ajalugu. Järgmises peatükis arutletakse veebi ligipääsetavuse kahe erineva vaatenurga nüansside üle: mida tähendab ligipääsetav veebileht kasutaja jaoks ja mis teeb selle probleemseks veebiarendaja jaoks. Kolmas peatükk keskendub EPIKojale arendatud veebilehel kasutatud tehnilistele lahendustele, nende valimise põhjustele ja teostamise keerukusele. Viimases peatükis analüüsitakse saadud tagasisidet ja kogutud andmeid ning hinnatakse valminud veebilehte nii ligipääsetavuse, jõudluse kui ka turvalisuse vaatenurkadest. Põgusalt tutvustatakse ka EPIKoja veebiarenduse tulevikus jätkamise plaane.

2 Ligipääsetavus

2.1 Ligipääsetavuse olemus ja tähtsus

Ümbritseva keskkonna ligipääsetavus mõjutab otseselt erivajadustega inimeste ühiskonnas elamise ning sotsiaalses, majanduslikus, poliitilises ja kultuurilises mõttes osalemise ning toimetuleku edukust. Analoogselt tavakeskkonnaga on vaja kohandada ka digitaalsed keskkonda ja seal pakutavaid teenuseid, et aidata inimestel seda hõlpsamini kasutada (Vandenbark, 2010). Lisaks tavakasutajatele on internetitarbijate hulgas palju neid, kelle kasutajakogemust halvendab tõsiselt mõni puue või näiteks vananemisega seotud võimekuse vähenemine.

Erivajadustega internetikasutajaid võib liigitada näiteks järgnevalt: pimedad, vaegnägijad, kurdid/vaegkuuljad, motoorikahäiretega inimesed, psüühikahäiretega inimesed (Koolmeister, 2016). Eesti ratifitseeris 2012. aastal ÜRO puuetega inimeste õiguste konventsiooni, võttes sellega endale kohustuse järjepidevalt erivajadustega inimeste olukorda parendada. Seega peab ka meie ühiskond nii lühikeses kui pikas perspektiivis lahendama probleeme, millega puuetega inimesed igapäevaelus kokku puutuvad.

Teine võimalus on erivajadused kolme laiemasse kategooriasse jaotada: kognitiivsed, sensoorsed ja motoorsed (Lazar, Dudley-Sponaule ja Greenidge, 2004). Samas ei arvesta kumbki jaotus näiteks keskkonnaga seotud erivajadustega, kus n-ö erivajaduseks võib olla elukoha infrastruktuurist tulenev ebapiisav internetiühenduse kiirus. Seega on väga lai hulk kriteeriumeid, millega tuleb veebikeskkonna loomisel arvestada.

Sierkowski (2002) defineerib veebi ligipääsetavust (edaspidi ligipääsetavus) kui kasutaja võimekust iga võimaliku riist- või tarkvaralise veebisisu kujutava abivahendi kaudu täielikult mõista ja kasutada lehekülje funktsionaalsust. Lihtsustatult võib ligipääsetavust kvantifitseerida UX-i kaudu ehk kui mugav ja intuitiivne on kasutajatel lehekülje teenuseid kasutada. Vaegnägijate puhul võib hinnata, kas veebileht pakub loetavuse parandamiseks võimalust sisu ümber paigutada ning tekstisuurust, kirjatüüpi ja/või kontrasti muuta (Okeke ja Izuogu, 2013).

2.2 Ligipääsetavuse ajaloost

Ligipääsetavus ei ole alati veebimaailma tähelepanu all olnud. 1990. aastate keskel ja lõpus toimusid ulatuslikud nn brauserisõjad, kus veebilehitsejad konkureerisid kasutajate arvu suurendamise nimel. Veebiarendajatele kujunes suureks probleemiks sellise sisu loomine, mis toimiks korrektselt iga veebilehitseja ja operatsioonisüsteemi kombinatsiooni puhul, seega ligipääsetavus oli teisejärguline. Sellel ajal oli suur osa diskussioonist suunatud “sujuva lagunemise” (ingl *graceful degradation*) põhimõttele, mis väitis, et sisu tuleks luua kõige arenenumate ja võimekamate veebilehitsejate jaoks ning teistsuguste ja vanemate tehnoloogiate kasutajate jaoks tuleks toota eraldi lahendus. Selle vastukaaluks tekkis uuenduslik “järk-järgulise tõhustamise” (ingl *progressive enhancement*) liikumine, mis defineeris konkreetsemalt veebilehe arendamise protsessi. Nimelt väitsid nad, et kõigepealt tuleks luua iseenesest kaasahaarav ning loogilise ülesehitusega sisu. Alles seejärel võib kaasajastada kujundust ja visuaalselt poolt, lisades modernsema disaini elemente. See tähendas, et veebiarendajad ei pidanud enam aega kulutama vananenud tehnika kasutajatele erilahenduste pakkumise peale (Baker, 2014).

Lisaks eelmainitud uuendustele toimus sellel ajal veebitehnoloogia vallas veel edasiminekuid. Sama kümnendi jooksul täiustati pidevalt nii veebisisu kujundamise keelt CSS kui ka veebisisu loomise keelt HTML, mille 4. väljalase sai 1998. aasta kevadel W3C ametlikuks soovitusel (Longman, 1998). Järgmisel aastal anti veebiarendajate ja erivajadustega veebikasutajate abistamiseks WAI raames välja WCAG 1.0 direktiiv – arendusprotsessi toetav reeglistik. Sellest hoolimata toob 2004. aastal tehtud uuring välja, et 21,7% veebilehtede haldajate sõnul pole nende veebilehed vaegnägijatele ligipääsetavad (Lazar, Dudley-Sponaule ja Greenidge, 2004).

Aastal 2017 tehti meta-analüüs, mis hõlmas aastatel 1999–2014 avaldatud 397 ligipääsetavust hindavat uurimust. Antud analüüsi käigus leiti järjepidevaid ilminguid ligipääsetavuse probleemidele, mis ei paista paranevat. Hoolimata mitmetest initsiatiividest, laialdaselt kättesaadavast ligipääsetavuse informatsioonist ja konkreetset protsessi struktuuri defineerivatest abinõudest ei ole arvestatav osa veebiarendajatest siiani suutelised ligipääsetavaid veebilehti looma (Swallow, 2017). Seega on tänapäeva veebikeskkondade ligipääsetavuse vallas arvukalt tähelepanu nõudvaid puudujääke.

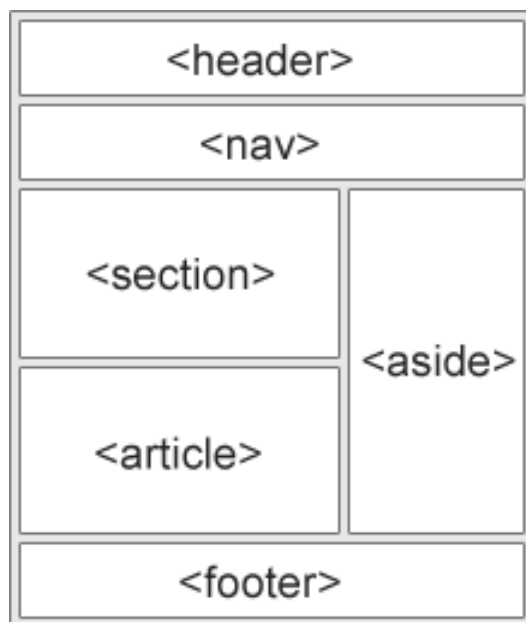
3. Ligipääsetavuse parendamise tehnilised ja regulatoorsed aspektid

Keskmine internetikasutaja pole tõenäoliselt ligipääsetavuse peale mõelnud – veebi probleemideta tarbides ei teki tal selleks vajadust. Levinum aruteluteema on hoopis kasutajakogemus, mis tava- ja erivajadustega kasutajate jaoks tähendab sama asja, kuid mille teostamine on erinev. Mugavalt töötav kasutajaliides tähendab rahulolevamaid ja seeläbi ka sagedamini tagasipöörduvaid kasutajaid. Olenevalt olukorrast on erivajadustega kasutajate jaoks hea UX erinev – sisu suurendamine, värvipaleti muutmine, kontrasti tõstmine, lehel mugavalt navigeerimine, sisu lugemine/kuulamine jpm. Lisafunktsionaalsus aitab samuti tavakasutajaid. Veebisisu luues on palju pealtnäha ebaolulisi ja tavapäraseid asju, kuid nende optimeerimisel paraneb nii tava- kui ka erivajadustega kasutajate UX märgatavalt. See on üks universaalse disaini peamisi põhimõtteid ning lihtsustab hiljem ka spetsiifilisemalt ligipääsetava disaini rakendamist.

3.1 Ekraanilugered

Ekraaniluger on tehnoloogiline abivahend, mis loeb ette veebilehel oleva sisuteksti ning ka pildid ja muu graafilise sisu, eeldusel, et viimased on korrektselt vormistatud. Seda kasutavad peamiselt inimesed, kelle nägemisvõimekus ei ole ekraanilt lugemiseks piisav: vaegnägijad ja pimedad. Üldiselt alustab ekraaniluger ette lugemist lehe ülemisest otsast navigatsiooniriba juurest ning liigub sealt järjest edasi. Navigeerimine käib klaviatuuri abil: *Tab* klahviga minnakse edasi järgmise elemendi juurde, *Shift-Tab* kombinatsiooniga saab naasta eelmise elemendi juurde (*University of Connecticut*, 2018).

Mõned paremad lugered pakuvad ka võimalust menüüsid vahele jätta ja minna otse asjakohase sisu juurde. Selle korrektse toimimise jaoks on tähtis kasutada õige semantilise nimetusega HTML-elemente, näiteks teksti puhul peab artikli pealkiri olema pealkirja tähistavate `<h1>` ja `</h1>` märgiste vahel. Veebilehe liigendamisel ja sisu paigutamisel on põhimõte analoogne: navigatsiooniriba jaoks on `<nav>` ja `</nav>` märgised, jaluse jaoks on `<footer>` ja `</footer>` märgised jne. Järgnevalt on märgistest koosneva struktuuri näitlikustamiseks toodud Joonis 3.1.



Joonis 3.1: Optimaalne veebilehe struktuur HTML-elementidega
(W3Schools 2020)

Ekraanilugeri abistavad lisaks semantilistele HTML-elementide nimetustele ka paar elemendi esimesele märgisele lisatavat atribuuti. Pildimaterjali puhul tasub välja tuua *alt* atribuut, mis kujutab endast pilti kirjeldavat teksti neile, kes pilti ei näe, sh ekraanilugeri kasutajad. See on kasulik ka tavakasutajale olukorras, kus pilt on vigane või internetikiirus ei võimalda pilti alla laadida.

Olukorras, kus eelmainitud HTML-elementide semantiliste nimetuste kasutamisel ja atribuutide lisamisel veebileht piisavalt ligipääsetav ei ole, saab kasutada spetsiaalseid W3C loodud *ARIA* atribuute (Faulkner, 2020). Need on mõeldud ainult ekraanilugerile lisainformatsiooni andmiseks. Ekraaniluger kasutab veebilehe kavandist ülevaate saamiseks veebilehe dokumendi struktuuri. Seda annab spetsiifilisemalt täpsutada *aria-role* atribuudiga, mis määrab elemendi rolli lehekülje struktuuris. Samuti saab kõigile elementidele lisada *aria-label* atribuudi, mis on mõeldud elemendile kirjeldava teksti juurde panemiseks. Tähtsat rolli mängib ka *aria-live* atribuut, mis ütleb ekraanilugerile, kuidas ja kuna antud elemendi teksti ette lugeda. Näiteks *aria-live="assertive"* märgib tähtsat infot, mis küll ei sega hetkel käsil oleva teksti lugemisele vahele, kuid suunab ekraanilugeri esimesel võimalusel selleni. See on kasulik näiteks olukorras, kus ekraanilugeri kasutaja on esitanud mingisuguse ankeedi ja andmete valideerimisel tekkis viga.

Ekraaniluger saab ülalmainitud märgistest ja atribuutidest aru ning kasutab neid, et lehe kasutamist võimalikult palju mugavamaks teha.

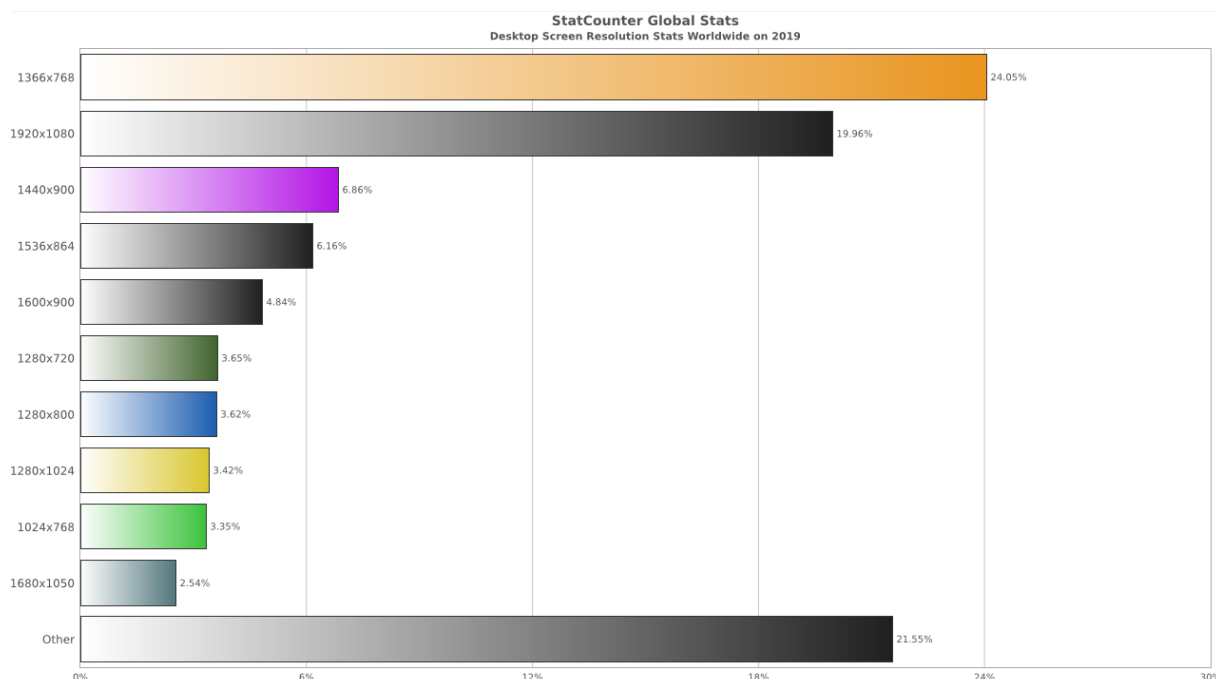
3.2 Veebisisu optimeerimine

Erinevate psüühikahäiretega inimeste jaoks (nt bipolaarne meeleoluhäire, skisofreenia) veebimaterjali luues tuleb arvesse võtta mitmeid erinevaid asjaolusid. Need on kokkuvõtlikult kirja pandud FEDM (ingl *Flat Explicit Design Model*) juhistes. Veebilehel ei tohi olla kogu vaba ruum infoga täidetud, elemendid peavad olema võimalikult eristatavad ja spetsiifilise teemaga seotud. Samuti tuleb jälgida, et ei kasutataks potentsiaalselt järsku meeleseisundi muutust põhjustavat sisu: segased animatsioonid, liikuvad hüplikaknad, taustamuusika, mida ei saa kinni panna jms. Abiks tuleb ka lihtsa lehekülje struktuuri ja alalehtede hierarhia rakendamine, kusjuures alalehele minnes võiks olla selgelt nähtavas kohas nn “leivapuru” rada (Rotondi et al., 2013).

Suurt rolli veebilehe kasutamise mugavuse juures mängib see, kui kiiresti otsitav info üles leitakse ning kui mugav on erinevate alalehtede vahel liikumine. Mingil määral teeb seda lihtsamaks nn “leivapuru” rada, aga paremaks lahenduseks on veebilehe sisuelementide optimaalne paigutamine. Pang et al. (2016) poolt läbi viidud uuringus salvestati inimeste pilgu liikumise teekonda erinevatel veebilehtedel ning töötati välja meetod olemasoleva veebilehe elementide suuruse, värvi ja paigutuse optimeerimiseks. Taaskord tuuakse välja varasemalt mainitud põhimõtted: kõige tähtsam ja asjakohasem informatsioon peab olema vastavalt esile toodud ning võimalikult lihtsasti kättesaadav ja märgatav.

Veebiteenuste laialdane kasutajaskond tähendab ka arvukalt erinevaid seadmeid, millega veebi tarbitakse. Olulist rolli veebiteenuste kujundamisel mängib kasutatavate seadmete ekraani eraldusvõime (ingl *resolution*). Statcounteri (2019) andmete põhjal moodustavad alla 1920x1080-pikslise eraldusvõimega kuvarid tarbijate poolt kasutatavate ekraanide seas üle 60%. Samuti saab andmetest järeldada, et ükski spetsiifiline eraldusvõime ei moodusta teiste ees selget enamust, seega tuleb veebi kujundades nende kõigiga arvestada.

WCAG 2.0 direktiiviga kooskõlas olemiseks peavad veebilehed olema disainitud skaleeruvalt alates ekraanilaiusest 320 px ning arvestatud peab olema vähemalt kuni 200% suurendusega. See on eriti oluline nägemispuudest tulenevate erivajadustega kasutajate puhul, kes võivad kasutada operatsioonisüsteemi või veebilehitsejasse sisseehitatud suurendust. Mitteskaaleeruva veebilehe puhul võivad sisuelemendid üksteisega kattuma hakata, mille tulemusena läheb osa informatsioonist kaotsi ja veebilehe kasutatavus väheneb. Levinumad kasutuselolevad lahutusvõimed on toodud Joonisel 3.2.



Joonis 3.2: Globaalselt kasutatavad lauaarvutite lahutusvõimed (StatCounter 2019)

Positiivne kasutajakogemus ei olene ainult pakutavast teenusest ja selle väljanägemisest, vaid ka teenuse kättesaadavusest – kui maailma kõige ligipääsetavama ja visuaalselt paeluvama veebilehe laadimine võtab liiga kaua aega, siis seda ei kasutata. Üks sekund kauem laadinud veebileht tähendab keskmiselt 7% võrra kahanenud ülemineku suhet. Seega Amazoni jaoks tähendaks ühe sekundi võrra kauem laadiv veebileht aastas 1,6 miljardi dollari võrra väiksemat müügitulu (Zuzana, 2016). Veebilehe laadimise kiirust mõjutavad väga paljud erinevad tegurid, mõned neist arendaja poolt mõjutatavad, mõned mitte. Arendaja saab sisu optimeerida nt pildifaile tihendades, pistikprogrammide ja ümbersuunamiste hulka vähendades ning ettelaadimise (ingl *prefetching*) tehnikaid kasutades. Taaskord aitab ühe kasutajagrupi UX parendamine ka teisi - väiksema mahuga pildimaterjal ja veebileht laeb oluliselt kiiremini ka kasutajate jaoks, kelle internetiühenduse kiirus on muidu ebapiisav.

3.3 Probleemid ligipääsetavusega

Power et al. (2012) viisid läbi empiirilise uuringu 64 erivajadusega veebikasutaja (32 pimedad, 19 osalise nägemispuudega ning 13 düsleksiaga) seas, kus analüüsiti kasutajate poolt kohatud probleeme erinevate WCAG 1.0 (Chisholm, Vanderheiden ja Jacobs, 1999) ja WCAG 2.0 direktiividele (Caldwell et al., 2008) vastavuse tasemetega veebilehtedel. Direktiividele vastavust kontrolliti manuaalsete auditite näol - juhendite nõuded käidi punkti kaupa läbi. Kasutajad külastasid 16 erinevat veebilehte, kus igal lehel lahendati teatud ülesandeid. Kokku tekkis kasutajatel 1383 erinevat probleemi mingisuguse ülesande lahendamisel. Paraku täheldati, et esinenud probleemidest oli kõigest 50,4% üldse direktiivis välja toodud. Seega võib järeldada, et WCAG 1.0 ja WCAG 2.0 juhendid ei kajasta olulisel määral erivajadustega kasutajate tegelikke probleeme. Samuti leiti, et antud probleemide lahendamiseks WCAG 2.0 poolt välja pakutud tehnikad ei ole optimaalsed – 16,7% eelmainitud veebilehtedest olid antud tehnikaid kasutanud, kuid need ei parandanud kasutajate kogemust.

Ülal mainitud uuring pakkus lõpusõnas välja ka potentsiaalse lahenduse: et veebiteenused ligipääsetavamad oleksid, tuleb liikuda probleemipõhisest lähenemisest hoopis disaini põhimõtetest johtuva lähenemise suunas. Lahenduse idee seisneb erivajadustega kasutajale veebilehekülje kujunduse üle suurema kontrolli andmises: kui kasutaja saab lehekülje kujundust oma soovile vastavalt muuta, on ta kasutajakogemus oluliselt parem.

3.3.1 WCAG 2.0 vs WCAG 2.1

WCAG versioon 2.0 on lähenemiselt eelmisest, WCAG 1.0 versioonist fundamentaalselt erinev, kuna ei keskenduta enam HTML'ga seotud ligipääsetavuse tehnikatele, vaid võimalikult paljudele erinevatele keeltele ja raamistikele. Arvestades veebitehnoloogia arengut, on see loogiline järgmine samm ligipääsetava veebisisu loomise juhendi koostamiseks. Kahjuks on sellisest laialdasest lähenemisest tekkinud arvukalt probleeme ligipääsetava veebi arendamisel (Sloan, 2016). Clark (2006) toob välja viis peamist WCAG 2.0 puudujääki:

1. Juhendis sisalduva materjali suur hulk, mis mõjutab negatiivselt elukutseliste arendajate ja disainerite kalduvust seda kasutada.
2. Juhendi nõuete ebamäärasus ja liigne üldisus tehnoloogilise neutraalsuse eesmärgil.

3. Tekkinud võimalus ebamääraselt sõnastatud nõudeid kuritarvitada direktiiviga võltsi vastavuse saavutamiseks.
4. Koodi valideerimise ja selle standarditele vastavuse kontrollimise võimaluse välja jätmine.
5. Kognitiivsete puuete ja õpiraskustega inimeste eripäradega tegelevate materjalide puudulikkus.

Alates versioonist 2.0 järgivad juhendis leiduvad nõuded ühte peamist printsiipi - veebileht peab olema tajutav, kasutatav, arusaadav ja robustne (ingl **P.O.U.R** – *Perceivable, Operable, Understandable, Robust*). Antud põhimõtet veebiarenduse käigus meeles pidades on võimalik kaine mõistuse ja loogikaga juba eos vältida mitmete ligipääsetavuse probleemide tekkimist.

Aastal 2016 hakati tegema eeltööd WCAG 3.0 loomiseks. Selleks, et avaldatud versioonide vahele ei jääks liiga pikka ajavahemikku, avaldati 2018. aastal WCAG 2.1. Selle peamised erinevused WCAG 2.0 versiooniga on eesmärk arendada ligipääsetavuse juhendeid kolme kasutajagrupi jaoks: kognitiivsete puuete ja õpiraskustega kasutajad, vaegnägijad ning mobiilseadmeid kasutavad erivajadustega inimesed. Antud direktiivi loomisega seotud töögrupp arendas, hindas ja täiustas arvukalt erinevaid viise, kuidas ülalmainitud kasutajagruppide vajadustega paremini arvestada. Lõpliku direktiivi versiooni kujunemisel mängis suurt rolli eelnevast versioonist tulenenud struktuurilised nõuded ning selle ülalmainitud puudujäägid. Valminud direktiiv täiustab järk-järgult veebisisu ligipääsetavust kolme kasutajagrupi vajaduste vallas, kuid kindlasti ei täida direktiiviga vastavuses olemine kõikide kasutajate vajadusi.

Kuna WCAG 2.1 loomisel on arvestatud eelneva versiooni struktuuriliste nõuetega, on see ühilduv WCAG 2.0 direktiiviga, mis tähendab, et WCAG 2.1 juhendiga vastavuses veebilehed on vastavuses ka eelmise versiooniga. Samuti saavad veebilehtede haldajad viia lehtede sisu vastavusse WCAG 2.1 juhendiga ilma eelneva versiooniga vastavust kaotamata.

3.3.2 Arendajate probleemide tagamaad

Mitme eelmainitud uuringu tulemused näitavad, et veebis on ligipääsetava sisuga selgelt probleeme. Selle põhjusteks on toodud arvukalt erinevaid probleeme. Swallow (2017) kategoriseerib põhjuseid järgnevalt:

- 1) keelelised ja terminoloogilised põhjused;

- 2) struktuursed ja organisatoorsed põhjused;
- 3) juhendite liigne maht ja kehv arusaadavus.

Keelelised ja terminoloogilised põhjused langevad kokku eelmainitud WCAG direktiivi ebamäärasuse ja liigse üldisusega. Samale järeldusele jõudis ka Okeke ja Izuogu (2013) uuring, kus leiti, et arendajate probleemide peamiseks põhjusteks on ligipääsetavuse materjalide ebapiisav spetsiifilisus ning jõustamise korda kehtestava seadusandluse puudulikkus.

Swallow (2017) lähenes probleemile sama uurimistöö raames ka inimesele ja nende käitumise põhjustele keskenduva vaatenurga alt, kasutades kontekstuaalse küsitlemise (ingl *contextual inquiry*) metoodikat 13 elukutselise veebiarendaja peal. Selle tulemusena sai ta välja tuua mitu huvitavat asjaolu.

1. Sageli on veebiarendajad motiveeritud ligipääsetavust arvestama konkreetse kohusetunde tõttu erivajadustega inimeste ees, isegi kui töö tellija pole seda nõudnud.
2. Mida suuremas organisatsioonis veebiarendaja töötab, seda vähem on tüüpiliselt klientidega otse suhtlemist ja seeläbi ka vähem arvestamist nende soovidega.
3. Kuigi arendajatel olid olemas ligipääsetavusega seotud algelised oskused, jäi nende teadmistest selgelt väheks.

Seega hoolimata arendajate tugevast sisemisest motivatsioonist pakkuda kõigile head kasutajakogemust, seavad nad tihti prioriteediks pigem programmeerimise head tavad ja standardid kui positiivse kasutajakogemuse, mis tuleneb ennekõike ebapiisavatest teadmistest erinevate ligipääsetavuse metoodikate osas ning juhendite ebamäärasusest.

4 Metoodika

Nagu sissejuhatuses mainitud, sai uus veebileht loodud koostöös teise üliõpilasega. Varasemalt ei olnud ühiselt tegeletud sellisel skaalal infotehnoloogia-teemaliste projektidega ega kasutatud eelmainitud meetodeid, seetõttu jäi igasuguse dokumentatsiooni kirjutamine tahaplaanile. Keskenduti peamiselt projekti tähtaegadele ja enda arendamisele kõikides veebiarendusega seotud valdkondades (sh ligipääsetavus, kujundus, funktsionaalsus) ning tehti tihedalt üksteisega koostööd. Välja saab tuua nii palju, et töö autor tegeles peamiselt Javascript'i ja PHP keerulisemate nüanssidega ning teine üliõpilane tegeles peamiselt EPIKojaga suhtlemise ning kujundusmallide kavandite loomisega. Kui kummalgi tekkis käsiloleva ülesande lahendamisel probleeme, lahendati enamasti see koos.

4.1 Tööprotsess

Esimeseks sammuks ligipääsetava veebilehe arenduses kujunes tööplaani paika panemine. Alustati diskussiooni EPIKojaga, et teada saada, millisel kujul ja millises mahus tuleks eelnevalt veebilehelt sisu üle tõsta ning kas ja milliseid funktsionaalsuseid tuleks juurde lisada. Seejärel otsiti erinevaid näiteid teadaolevatest ligipääsetavatest veebilehtedest, kust saadi inspiratsiooni mitme tehnilise lahenduse teostamiseks (nt link "Liigu põhisisu juurde"). Järgmisena vaadati põhjalikult läbi WCAG 2.0 direktiiv, et leida olulisemad ligipääsetavust parandavad nõuded. Nõuded, mis olid direktiivis välja toodud, kuid mille teostamiseks tehnilist lahendust pakutud polnud, täideti suures osas ise kirjutatud koodiga. Ülejäänud nõuete tehniliste lahenduste teostamiseks saadi inspiratsiooni keskkonnast *codepen.io*, kust leitud koodi oma vajadustele vastavalt edasi arendati. *Codepen* on veebipõhine avatud arenduskeskkond, kus veebiarendajad saavad luua, testida ja jagada oma loodud lahendusi erinevate eesmärkide täitmiseks.

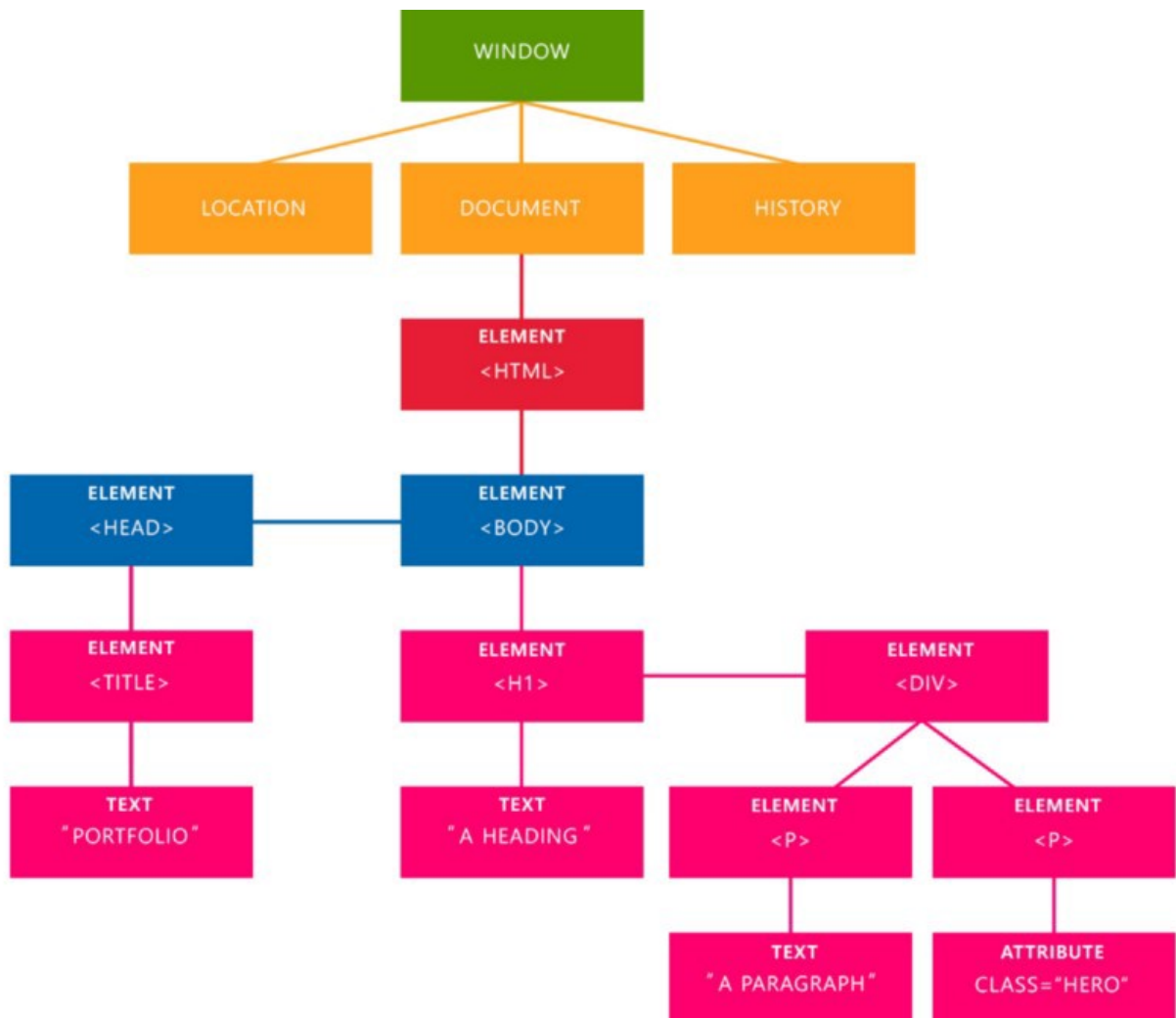
Üks EPIKoja märkimisväärsitest soovidest oli erivajadustega inimeste kujutamine illustratsioonides mitte ratastoolis, vaid mingil muul, neutraalsemal kujul. Selle tegi oluliselt lihtsamaks *unDraw.co* keskkond, kus on arvukalt erinevaid tasuta ja kohandatava värvipaletiga illustratsioone. Selle keskkonna valimise kasuks rääkis ka asjaolu, et EPIKoja veebilehe haldajad saavad sealt tulevikus vajadusel ise uusi illustratsioone valida ning lehele lisada.

Veebilehe majutamise jaoks teenusepakkuja valimisel probleeme ei tekkinud – eelnev veebileht oli Eestis paiknevas *zone.ee* veebiserveris, mis kattis täielikult veebilehe kõik vajadused, seega kasutati seda ka uuendatud veebilehe majutamiseks. Kui arutelu käigus oleks jõutud järeldusele, et on vaja uut serverit, oleks valitud teine Eesti teenusepakkuja – *veebimajutus.ee* - mis on autori varasema kogemuse põhjal samuti tehniliselt võimekas ning operatiivse kasutajatoega.

4.2 Kasutatud keeled

HTML kirjeldab veebilehitsejale lehekülje struktuuri ja kuidas seda kuvada ning koosneb elementidest, mis on ümbritsetud $\langle \rangle$ märgistega, nt `<div> elemendi sisu </div>`. Nendest elementidest saavad veebilehe alustalad, kusjuures märgiseid lõppkasutajale ei näidata, vaid neid kasutatakse lehekülje sisu kuvamiseks.

Veebilehitseja loob endale HTML'i poolt kirjeldatud sisust dokumendi objektide mudeli (DOM, ingl *Document Object Model*). Selle modifitseerimise kaudu toimuvad enamik lehekülje elemente ja nende sisu muutvatest protsessidest veebilehel, kusjuures veebilehte uuesti laadimata. Samal meetodil toimib ka jQuery kaudu sündmuste käitlemine. Joonisel 4.1 on näha lihtsakoelise veebilehe järgi loodud dokumendi objektide mudel.



Joonis 4.1: Veebilehitseja loodud DOM (Maldonado, 2018)

4.2.1 Struktuur ja funktsionaalsus

Allpool on toodud kaks koodinäidet vastavalt HTML'i ja PHP sünergiast ning eraldiseisva PHP töötamise loogikast.

Joonisel 4.2 on osa Kirby CMS kujundusmalli koodist, kus on võimalik tekitada veebilehe struktuur HTML-elementidest ning nende sisu dünaamiliselt genereerida Kirby halduspaneelist PHP kaudu saadud kasutajasisendi põhjal.

```

<div class="tab-wrapper">
    <ul class="tab-menu">
        <?php foreach($page->children() as $subpage): ?>
            <li><?= html($subpage->pealkiri()) ?></li>
        <?php endforeach ?>
    </ul>
    <div class="tab-content">
        <?php foreach($page->children() as $subpage): ?>
            <div>
                <?= $subpage->dokumendid()->kt() ?>
                <?= $subpage->dokumendideditor()->blocks() ?>
            </div>
        <?php endforeach ?>
    </div>
</div>

```

Joonis 4.2: Kirby CMS kujundusmalli kood

Järgneval lehel (vt. Joonis 4.3) on toodud näide PHP keeles kirjutatud EPIKoja veebilehe otsingumootori töötamise loogikast. Kirby sisuhaldustarkvara laialdase dokumentatsiooni ning sisseehitatud funktsionaalsuse tõttu on tulemuseks saadud kood üpris loetav ja kokkuvõtlik. Otsingufunktsioon töötab veebilehel kui globaalne API, mille tagastatavaid muutujaid saab koodis kasutada.

```

return function ($site) {

    $query = get('q');
    $results = $site->index()->search($query, 'title|text|tags|sisu|pealkiri');
    $results = $results->paginate(20);

    return [
        'query'    => $query,
        'results' => $results,
        'pagination' => $results->pagination()
    ];
};

```

Joonis 4.3: EPIKoja veebilehe otsingumootori kood

4.2.2 Kujundus

CSS (ingl *Cascading Style Sheets*) – keel, mida kasutatakse veebilehtede ja -rakenduste sisu paigutamiseks ning visuaali kujundamiseks. Seda kasutades on võimalik muuta HTML-elementide asukohta, kuju, mõõtmeid, värvi, luua primitiivseid animatsioone jpm.

CSS ei kujuta endast standardset programmeerimiskeelt, vaid selle ülesanne on pigem paika panna erinevate elementide omadused, mida on kindlaksmääratud arv. Mitmele elemendile samaaegselt reeglite määramiseks on kasutusel klassid (ingl *class*), mida saab lisada kuitahes mitmele HTML-elementile *class* atribuudiks. Selle tulemusena kehtivad neile kõigile klassis määratud väärtustega reeglid. Olukorras, kus on vaja reegleid rakendada vaid mõnele spetsiifilisele elemendile paljudest, saab kasutada *id* atribuuti. Vastavale *id* atribuudile saab analoogselt määrata kindlad reeglid, mis seejärel kehtivad vaid sellega seotud HTML-elementidele.

CSS2 (välja lastud aastal 1998) tõi endaga kaasa uue *@media* reegli, mis võimaldab kehtestada erinevate meediumi tüüpide puhul isesugused kujundusreeglid. CSS3 arendas antud ideed edasi, lisades aastal 2012 uue *media-query* nime kandva funktsionaalsuse. Sellega saab vastavalt kasutatava seadme omadustele muuta veebilehe kujundust. Näiteks saab paika panna reeglid, mis kehtivad ainult olukorras, kus kasutaja ekraani laius, ja nutiseadmete puhul

ka ekraani pikslitihedus, on arendaja poolt kehtestatud vahemikus. See võimaldab luua veebilehti, mis on täielikult skaleeruvad iga seadme jaoks.

Järgnevalt on toodud näide CSS koodist (vt. Joonis 4.4), mis määrab kujundusreeglid uudiskirja HTML-elementidele olukorras, kus kasutaja ekraani laius on minimaalselt 1700 px.

```
@media screen and (min-width: 1700px) {  
    .more-container .more-item,  
    .more-container .gap {  
        width: 23.5%;  
    }  
    #more-uudiskiri {  
        width: 23.5%;  
        height: 100%;  
        margin-top: 0px;  
    }  
    .uudiskiri {  
        padding-top: 33px;  
        padding-bottom: 13px;  
    }  
    .uudiskiri-pilt {  
        display: none;  
        visibility: hidden;  
    }  
}
```

Joonis 4.4: CSS kujundusreeglid *media-query* näitel

4.3 Kasutatud pistikprogrammid, teegid

Siin peatükis tutvustatakse erinevaid teke ja pistikprogramme, mis tegid arendamisprotsessi oluliselt hõlpsamaks. Antud käsitluses on pistikprogramm väline funktsionaalsuste kogumik, mille initsialiseerimiseks koodiga tegelema ei pea. Teekide all on mõeldud vastava keele standardsetele funktsioonidele lisaväärtust pakkuvate funktsioonide kogumikke, mida on võimalik paigaldamise järel koodi kirjutamisel kasutada. Kõikide kasutatud teekide ja pistikprogrammide viited leiab käesoleva bakalaureusetöö kasutatud allikate nimekirjast.

4.3.1 jQuery

jQuery on üks paljudest JavaScripti teekidest, mis teeb mugavamaks erinevatele HTML-elementidele ligi pääsemist ja nende atribuutide muutmist. Samuti lihtsustab selle võimekus sündmuste käitlemist (ingl *event handling*), võimaldades neid kergelt mingi ajalise intervalli tagant käivituma panna. jQuery teegi kasutamine on üpris lihtsakoeline – selleks on vaja dollarimärki (\$), misjärel saab kasutada teegi funktsioone. Samuti on dollarimärgiga tähistatud jQuery abil loodud muutujad, mida saab täielikult integreerida JavaScript'i koodis. EPIKoja veebilehe arenduses kasutati jQuery't (versioon 4.3.1) peamiselt HTML-elementide atribuutide muutmiseks mingisuguse kasutaja põhjustatud sündmuse tulemusel, näiteks veebilehe kirjatüübi suurendamise nupul klikkides. jQuery'l on ka väiksemaid lisateeke, millest kasutati jQuery Validate'i (versioon 1.19.1). See võimaldab hõlpsalt ühe funktsiooniga valideerida ankeetide sisendeid.

Joonisel 4.5 on toodud näide jQuery teegi abil HTML-elementidele ligi pääsemisest, nende atribuutide muutmisest ning mõnest teegi sisseehitatud funktsioonist (nt *find*, *attr*, *not*, *each*, *filter*).


```

$(".tab-menu li:first").addClass("active");
var $wrapper = $('.tab-wrapper'),
    $allTabs = $wrapper.find('.tab-content > div'),
    $tabMenu = $wrapper.find('.tab-menu li'),
    $line = $('<div class="line"></div>').appendTo($tabMenu);

$allTabs.not(':first-of-type').hide();
$tabMenu.filter(':first-of-type').find(':first').width('100%')

$tabMenu.each(function(i) {
    $(this).attr('data-tab', 'tab'+i);
});

$allTabs.each(function(i) {
    $(this).attr('data-tab', 'tab'+i);
});

```

Joonis 4.5: jQuery teegi funktsionaalsus

4.3.2 Matomo Web Analytics

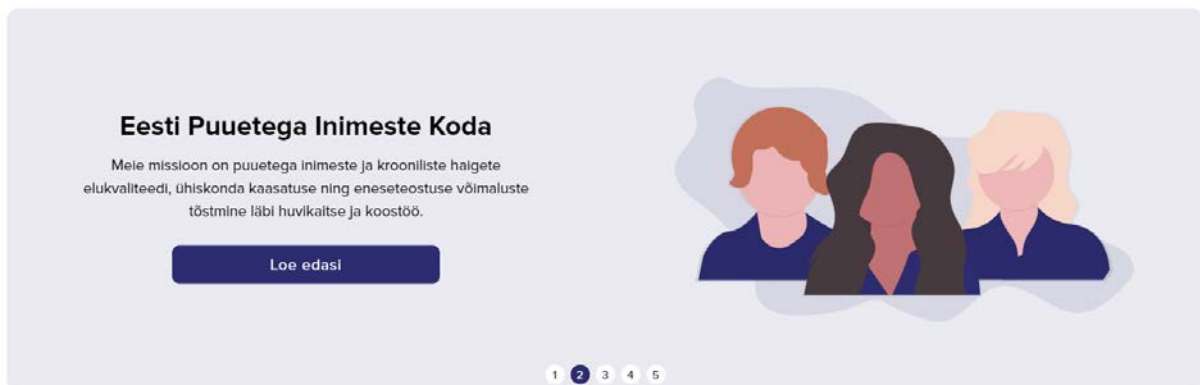
Matomo Web Analytics kujutab endast JavaScriptis kirjutatud pistikprogrammi, mis pakub väga mitmekülgset veebilehe kasutajate ja muude näitajatega seotud analüütika teenust. Google Analytics asemel valiti Matomo Web Analytics (versioon 3.13.4), kuna selle tasuta versioon pakub võrreldes esimesega rohkem funktsionaalsust ja on avatud lähtekoodiga – see tähendab, et potentsiaalsed tulevikus tekkivad kohandamisvajadused ei nõua kogu analüütika mooduli välja vahetamist. Allpool on välja toodud peamised Matomo pakutavad võimalused, mille tõttu antud analüütika moodul valiti.

1. Andmete jälgimiseks ja haldamiseks mõeldud paneeli täielik kohandatavus.
2. Veebilehe külastajate andmete reaajas jälgimise võimalus.
3. Veebilehe otsingumootori otsingute salvestamise võimalus.

4. Veebilehe külastaja teekonna detailse jälgimise võimalus, seeläbi saab analüüsida, milline lehe osa toob sisse kõige rohkem külastajaid, kust kaudu nad sinna on jõudnud ning millisel alalehel kasutajad kõige enam lahkuvad.

4.3.3 Swiper-js

Swiper-js on JavaScripti teek, mis lihtsustab oluliselt karusselli (ingl *carousel*) stiilis elementide loomist. Karussell kujutab endast praktiliselt pildigaleriid, kus korraga kuvatakse ühte slaidi, mis üldiselt vahetub järgmise vastu kindla ajavahemiku tagant. Swiper-js teek on tarkvarakiirenduse võimekusega, avatud lähtekoodiga ning täiesti tasuta. Swiperi teegi kasutamine välistab täielikult vajaduse JavaScript'is karusselli slaidide vahetamist manuaalselt rakendada ja on väga laialdaselt kohandatav. EPIKoja veebilehel kasutati selle teegi (versioon 5.3.8) funktsionaalsust avalehel oleva karusselli loomiseks. Swiper-js abil sai karussellile mugavalt lisatud ka hetkel aktiivset slaidi ja nende vahetamist võimaldavad numbrid (vt. Joonis 4.6).



Joonis 4.6: EPIKoja avalehel olev karussell

Järgneval lehel on toodud näide swiper-js teegi karusselli initsialiseerimise funktsioonist (vt. Joonis 4.7). Seal määratakse ära karusselli peamised omadused, nimelt slaidide vahetumise intervall, klaviatuuriga navigeeritavus ning kas karussell käib ringiratast või mitte. Samuti defineeritakse slaidide ja navigatsiooninuppude CSS'i klassid, mille abil saab hiljem nende kujundust vastavalt vajadustele muuta.

```

var swiper = new Swiper('.swiper-container', {
  loop: true,
  grabCursor: true,
  keyboard: {
    enabled: true,
  },
  autoplay: {
    delay: 15000,
    disableOnInteraction: true,
  },
  navigation: {
    nextEl: '.swiper-button-next',
    prevEl: '.swiper-button-prev',
  },
  pagination: {
    el: '.swiper-pagination',
    clickable: true,
    renderBullet: function (index, className) {
      return '<span class="' + className + '">' + (index + 1) + '</span>';
    },
  },
});

```

Joonis 4.7: Swiper-js teegi abil karusselli initialsiseerimine

4.3.4 Küpsised ja js-cookie

Küpsised (ingl *web cookie, browser cookie*) – veebilehtedel kasutatav väike andmepakett, mis salvestatakse kasutaja veebilehitsejasse. Tavaliselt talletatakse sinna kasutaja sessiooniga seotud andmeid ja veebilehel tehtud valikuid. Küpsiste kasutamist võib vaadelda kui meetodit UX parendamiseks, nt autentimise küpsis (ingl *authentication cookie*) jätab meelde, et kasutaja on sisse loginud ning veebilehe värskendamisel või peale sulgemist naasmisel ei pea kasutaja uuesti ennast autentima. Teek js-cookie defineerib kergelt kasutatavad funktsioonid, millega saab luua kindla sisuga küpsiseid ning antud küpsiste sisu lugeda.

EPIKoja veebilehel kasutati järgnevaid küpsiseid kasutaja mugavuse huvides:

- 1) küpsised kasutajate valikute kohta veebilehel kuvatava tekstisuuruse, värvipaleti (kontrasti muutmise eesmärgil) ja kirjatüübi osas;
- 2) küpsis, mis talletab kasutaja valiku EPIKoja uudiskirjaga liitumise kohta - järgnevatel külastuskordadel ei kuvata enam samale kasutajale liitumise hüpinkakent;
- 3) veebiliikluse analüütika pistikprogrammi Matomo küpsis;
- 4) Kirby sisuhaldustarkvara sisseehitatud autentimisküpsis.

Ligipääsetavusega seotud seadeid kontrollivate küpsiste kestuseks on määratud 14 päeva, uudiskirja küpsise kestuseks 365 päeva. Antud küpsiste loomiseks ja lugemiseks on kasutatud js-cookie teeki (versioon 2.2.0).

Joonisel 4.8 on toodud js-cookie teegi abil küpsise loomiseks kasutatud kood. Antud näites on kirjeldatud funktsiooni, mis kirjatüübi vahetamisel küpsisesse salvestatud numbrit suurendab.

```
var advanceFontIndex = function() {  
    if (fontTypeIndex < fontTypeMaxIndex) {  
        fontTypeIndex++;  
        Cookies.set("font", fontTypeIndex, { expires: 14});  
    } else {  
        fontTypeIndex = 0;  
        Cookies.set("font", fontTypeIndex, { expires: 14});  
    }  
}
```

Joonis 4.8: js-cookie teegi abil küpsise kasutamine

4.3.5 Nested Accordion

Nested Accordion on JavaScripti teek, mis võimaldab hõlpsasti erinevaid tekstiplokke dünaamiliselt kuvada ja peita. Seeläbi saab ühel alalehel kujutada märgatavalt rohkem informatsiooni kasutajale võimalikult loetaval ja optimaalselt liigendatud kujul.

EPIKoja arenduses kasutati teegi versiooni 1.4.7.3. Järgnevalt on toodud kuvatõmmis EPIKoja Õigusnõustamise alalehelt (vt. Joonis 4.9), mis kujutab antud teegi funktsionaalsust kasutades akordioni stiilis tekstiplokkide liigendamist. Sinisele kolmnurgale vajutades saab valida vastava teemaploki, misjärel vahetub kuvatav tekst antud teemaploki sisuks.

Uuri lisaks:

▼ Nõustamine

► Õigusnõustamine

► Nõustamine töövõime või puude raskusastme hindamise taotlejatele

Kui sa vajad abi või nõustamist puude ja erivajadusega seonduvates igapäeva elu puudutavates küsimustes, siis võta meiega **ühendust**. Jagame vajalikke infot ja kontakte ning püüame leida üheskoos vastuseid kõikvõimalike teemade, küsimuste ja murekohtade osas, nt õigused, õppimine, töötamine, iseseisev elu, teenused, toetused jpm.

Samuti võta julgesti **ühendust**, kui vajad koolitusi ja nõustamist. Lisaks pakume konsultatsioone, aitame koostada teabe- ja infomaterjale, lisaks koostame ja viime läbi täiskasvanute täiendkoolitusi nii oma liikmesorganisatsioonidele kui ka koostööpartneritele.

Joonis 4.9: Teksti liigendamine EPIKoja veebilehel Nested Accordion teegi abil

4.4 Tähtsamad täidetud WCAG nõuded

Visuaalselt arusaadava kasutajaliidese loomiseks veebis tuleb tagada:

- 1) sobiv kirjatüüp ja selle suurus;
- 2) piisav rea- ja lõiguvahe;
- 3) piisav kontrast sisuelementide ning sisuteksti ja tausta vahel;
- 4) elementide loogiline paigutus veebilehel.

Kasik (2007) seletab, et tüpograafia peamiseks reegliks on, et mida tuttavam tüpograafia ja tekstikuju, seda kergem on antud materjali lugeda. Loetavuse seisukohalt on tähtis ühtse stiili kasutamine (üks kirjatüüp kogu materjali peale) ja rahulikkus – pigem üks suur pilt kui mitu väikest. Sama võib väita ka pealkirjade kohta: kui igale lõigule pealkiri panna, väheneb nende

infoväärtus. Teksti parema loetavuse huvides on mõistlik kasutada väiketähti, mille struktuur on iseloomulikum ja neid on kergem eristada. Osal kirjatüüpidest esinevad tähemärke kaunistavad jalakesed/kriipsud ehk seriifid, mis küll paberkandjal hõlbustavad teksti lugemist, kuid ekraanilt on loetavamad ilma seriifideta ehk *sans-serif* kirjatüübid.

4.4.1 Success Criterion 1.1.1 Non-text Content

Mittetekstilise sisu (ingl *non-text content*) all mõeldakse igasugust sisu, mis ei ole tähemärkide jada või mis ei väljenda inimkeeles mingit mõtet (Cooper, 2018). WCAG direktiivi nõuetele vastamiseks peab olema veebilehe igasugusel mittetekstilisel sisul tekstiline alternatiiv. Pildimaterjali puhul on selleks `` märgise *alt* atribuut, mis kuvab tekstilise alternatiivi ekraanilugeri kasutajatele või tavakasutajatele olukorras, kus pilti korrektselt ei näidata. EPIKoja veebilehe haldajatele on mugavuse huvides tekitatud võimalus pildimaterjalile *alt* atribuuti lisada juba haldamise paneelis.

4.4.2 Success Criterion 1.3.5 Identify Input Purpose

Paljud veebilehed kasutavad kasutajate registreerimiseks, meililistiga liitumiseks jms kasutaja poolt täidetavaid ankeete. Nende ligipääsetavaks kohandamine on väga oluline, sest algelises vormis ankeete on ekraanilugeri kasutajatel peaaegu võimatu kasutada. Ankeet koosneb erinevast arvust sisendiväljadest `<input>` (nt nimi, meiliaadress jpm), mis tavakasutaja jaoks on märgistatud kohatäiteks (ingl *placeholder*) mõeldud tekstiga, mis kaob sisendiväljale vajutades. Selle ekraanilugeri ligipääsetavaks tegemiseks tuleb iga sisendivälja juures selgesõnaliselt välja tuua, mis sinna kirjutada tuleb ja kas see on ankeedi saatmiseks nõutud.

Sisendivälja ja sellega seotud nimesildi jaoks on mõeldud kaks HTML-elementi, vastavalt `<input>` ja `<label>` märgistega elemendid. Sisendiväljale tuleb lisada *type* ja *id* atribuudid, mis vastavalt määravad ära sisendiväljale sisestatava teksti eesmärgi (nt meiliaadress, telefoninumber) ja seovad selle kindla nimesildiga. Nimesildile tuleb lisada *for* atribuut, mis peab kattuma sisendivälja *id* atribuudiga.

4.4.3 Success Criterion 1.4.3 Contrast

Kontrast kirjeldab kui lihtne on kasutajal eristada kas teksti taustavärvist või ühte sisuelementi teisest kõrvalasetsevast sisuelemendist eristada, mis oleneb suuresti värvikontrastist. Kahe

värvi vahelist kontrasti saab mõõta kasutades värvikontrasti analüsaatoreid, mis on veebis vabalt kättesaadavad.

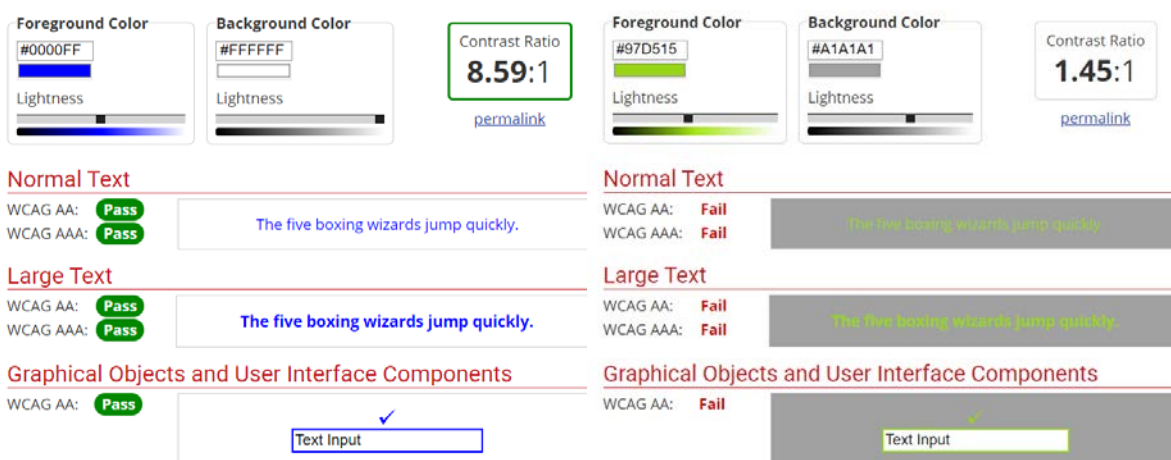
WCAG 2.0 direktiiv on paika pannud vastuvõetavad piirmäärad kontrastile:

- a. normaalsuuruses tekstile ehk 18 pt tavakirjas ja 14 pt paksus kirjas **või vähem**:
 - AA tase: suhe 4.5:1
 - AAA tase: suhe 7:1
- b. suurele tekstile ehk 18 pt tavakirjas ja 14 pt paksus kirjas **või suurem**:
 - AA tase: suhe 3:1
 - AAA tase: suhe 4.5:1

AA taseme nõuetele vastamine tagab loetavuse madala nägemisteravuse, värvipimeduse ja vananemisest tingitud loomuliku kontrastitaju vähenemise puhul enamikul juhtudest. Kontrasti suhe alla AA taseme ei ole WCAG reeglistiku järgi sobilik veebisisu jaoks. EPIKoja veebilehel on kasutusel kas valge tekst tumedal taustal või must tekst heledal taustal, kummagi olukorra värvikontrast vastavalt **12.27:1** ja **17.53:1**. Värvide kontrasti mõõtmiseks kasutati WebAIM värvikontrasti analüsaatorit. Kasutatud värvid veebivärvide kujutamiseks kasutatavas heksadetsimaal- ehk kuueteistkümnendsüsteemis (ingl *hex code*):

- a. Musta teksti jaoks kasutatud taustavärv - **#EAEAF1**
- b. Valge teksti jaoks kasutatud taustavärv - **#2D2C71**

Veebilehe värvipaleti/-teema arendamisel lähtuti olemasolevast EPIKoja brändist ning ülalmainitud kontrastinõuetest. Sobivat ja mitesobivat kontrasti demonstreerib Joonis 4.10.

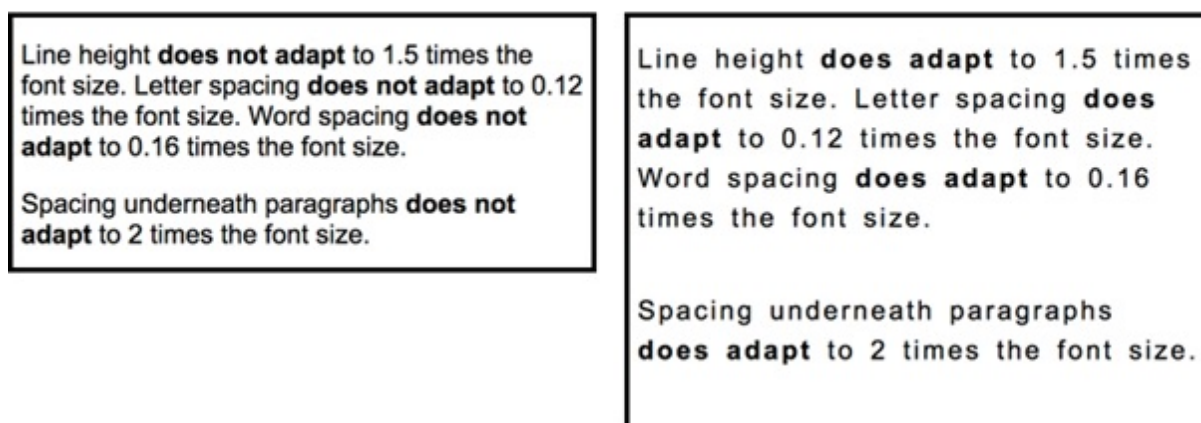


Joonis 4.10: Näide sobivast ja mitesobivast värvikontrastist (WebAIM, 2020)

4.4.4 Success Criterion 1.4.12 Text Spacing

WCAG 2.1 järgi on defineeritud minimaalne reavahe: 1.5 korda kirjatüübi suurus; minimaalne lõikude vahe: 2 korda kirjatüübi suurus; minimaalne tähemärkide vahe: 0.12 korda kirjatüübi suurus; ja minimaalne sõnade vahe: 0.16 korda kirjatüübi suurus. Nendele nõuetele vastamine tagab teksti minimaalse loetavuse. Joonisel 4.11 on toodud juhise paremini visualiseerimiseks pilt W3C *GitHub*'i lehelt. Parempoolne kast demonstreerib, kuidas käitub korrektselt vormistatud tekst kirjatüübi suuruse muutmisel, vasakpoolne kast kujutab nõuetele mittevastavat olukorda.

Tekstiga seotud nõuete jälgimine mängis suuremat rolli uudiste ja teiste tekstilise sisuga alalehtede paigutuse korraldamisel.



Joonis 4.11: Korrektselt vormistatud teksti dünaamika (Kirkpatrick, 2018)

4.4.5 Success Criterion 2.1.3 Keyboard (No Exception)

Ekraanilugerit tutvustavas peatükis toodi välja, et nende kasutajad navigeerivad veebilehel klaviatuuri abil – selle optimaalseks töötamiseks tuleb veebilehe elemendid järjestada loogiliselt ning välistada olukorrad, kus mingisuguses kohas jääb kasutaja nõ löksu (ingl *keyboard trap*), kust ei ole võimalik klaviatuuri abil edasise sisuni jõuda. WCAG direktiivi nõue lisab sellele veel järgneva: kogu veebilehe funktsionaalsusele ligi pääsemiseks ei tohi olla vajalik individuaalsete klahvivajutuste spetsiifiline ajastus. EPIKoja veebilehe kogu funktsionaalsus on ulatuslikult klaviatuuriga testitud ning nii veebileht kui selle haldamiseks

kasutatav paneel on mõlemad vastavalt WCAG nõuetele täielikult klaviatuuriga navigeeritavad.

Lisaks välja toodud nõuetele on omajagu väiksemaid, erivajadustega kasutajate UX-i parandavaid funktsionaalsuseid. Nendest mõned tuuakse järgnevas lõigus välja koos kasutuselevõtmise põhjustega.

Ekraanilugeri kasutajatele on veebilehe ülemises otsas, navigatsiooniribal peidetud link “Liigu põhisisu juurde“, mis aitab kasutajal veebilehe külastamisel vältida iga kord menüüdest läbi liikumist ning viib fookuse otse põhisisu (nt uudise artikkel) peale. EPIKoja lehel on lisaks navigatsiooniribale eraldi ligipääsetavuse ja kiirfunktsioonide riba, mis koosneb viiest funktsionaalsuse järgi jagunevast sektsioonist.

Esimene osa on kirjatüübi muutmise nupp, mis vahetab veebilehel kasutatavat kirjatüüpi kolme eelnevalt defineeritud tüpograafiliselt erineva kirjatüübi vahel. Vastavalt tüpograafia peamisele põhimõttele (Kasik, 2007) on tüpograafiliselt ja tekstikujult tuttavamat teksti kergem lugeda, selle funktsiooni implementeerimisel saab kasutaja valida endale loetavaima kirjatüübi.

Järgmine sektsioon on kirjatüübi suurendamise ja vähendamise nupud, mis teevad kogu veebilehe tekstilist sisu vastavalt suuremaks ja väiksemaks. Selle rakendamise tagamaaks on eesmärk pakkuda lisaks veebilehitseja ja operatsioonisüsteemi suurendusele paindlikumat funktsionaalsust.

Kolmas osa kujutab endast kogu veebilehe värvipaleti muutmise nuppu, mille eesmärk on tagada parim võimalik kontrast erinevate elementide vahel ja seeläbi kõrgeim loetavus olukorras, kus tavasätetest ei piisa. Valikus on kolm erinevat versiooni: tavasäte, kus veebileht kasutab EPIKoja brändingust tulenevat värvipaletti, ning kaks erinevat kõrgema kontrastiga sätet, kus mustal taustal on kasutatud vastavalt kollast ja valget teksti.

Neljas sektsioon on võrdlemisi lihtsakoeline – see taastab kõik ligipääsetavuse parandamise eesmärgil tehtud muudatused (nt kirjatüübi muutmine, teksti suurendamine/vähendamine, värvipaleti muutmine) tagasi algsätete peale. Viies osa on kiirotsingu nupp, mis viib kasutaja veebilehe otsingumootori alalehele.

4.5 Sisuhaldustarkvarad

Sisuhaldustarkvara (CMS, ingl *Content Management System*) kujutab endast veebisisu haldamise tarkvara, mis võimaldab kasutajal mugavamalt sisu luua, hallata ja muuta. Mõned neist välistavad täielikult vajaduse koodi kirjutamise tehniliste teadmiste järele, lastes kasutajal keskenduda rohkem kujundamisele (Kinsta, 2020).

Sisuhaldustarkvarade tüüpe on arvukalt ning neil kõigil on oma kindlad eelised ja puudused. Kliendi vajaduste järgi veebilehe jõudluse, turvalisuse, portatiivsuse ja paindlikkuse osas saab valida igaks olukorraks sobivaima sisuhaldustarkvara. Järgnevalt tutvustatakse paari levinumat CMS'i tüüpi.

Traditsiooniline, andmebaasil põhinev sisuhaldustarkvara hõlmab endas mitut erineva funktsionaalsusega komponenti, millest väärivad nimetamist sisu haldamise rakendus (ingl CMA - *content management application*), sisu edastamise rakendus (ingl CDA - *content delivery application*) ja andmebaasi haldussüsteem (ingl DBMS – *database management system*). Sisu haldamise rakenduses toimub sisu loomine ja muutmine, misjärel sisu edastamise rakendus teeb kasutaja eest ära igasugused kasutajapoolse sisendi salvestamise, ladustamise ja kuvamisega seotud toimingud. Traditsioonilise CMS'i peamised eelised teiste ees on haldamise lihtsus kasutajate jaoks, kellel pole põhjalikke teadmisi veebiarenduse tehnilisest osast ning populaarsuse tõttu ka saadavalolevate pistikprogrammide hulk (Kinsta 2020).

Failipõhine sisuhaldustarkvara (ingl *file-based CMS*, *flat-file CMS*) eristub selle poolest, et sellega ei ole seotud andmebaasi. Kogu veebilehe struktuur peegeldub kaustade ja failide struktuuris, kogu sisu on salvestatud tekstifailides ning selliselt on sisuhaldustarkvara mahult väiksem. Sellisel lahendusel on ka parem jõudlus, kuna ei ole andmebaasi, kuhu oleks vaja teha päringuid. Struktuurpäringukeele kaudu tehtud rünnakud (ingl *SQL injections*) on peamine viis veebilehtede kallale tungimiseks (Rogojan, 2020). Sellel põhjusel räägib andmebaasi puudumine failipõhise sisuhaldustarkvara eelistamise kasuks – seekaudu on veebilehel vähem haavatavaid osasid. Failipõhiste sisuhaldustarkvarade peamine eelis teiste tüüpide ees on siiski portatiivsus – kuna terve veebilehe struktuur ja sisu on failide ning kaustade kujul, on seda oluliselt hõlpsam ringi tõsta ja vajadusel mujal kasutusele võtta.

API-põhine sisuhaldustarkvara (ingl *API-driven CMS, headless CMS*) on kõige uuem ja modulaarsem CMS tüüp, mis sisaldab endas sarnaselt traditsioonilisele CMS'le sisu hoiustamise ja haldamise osa (CMA) ning sisu kohale toimetamise osa (API). Selline lähenemine võimaldab arendajal API kaudu loodud sisu kasutada igal pool, mitte ainult sisu hoiustamise kohas. Sellise lähenemise suurim eelis on võimalus veebilehe sisu integreerida igasuguse keele ja/või raamistiku abil kirjutatud keskkonnas.

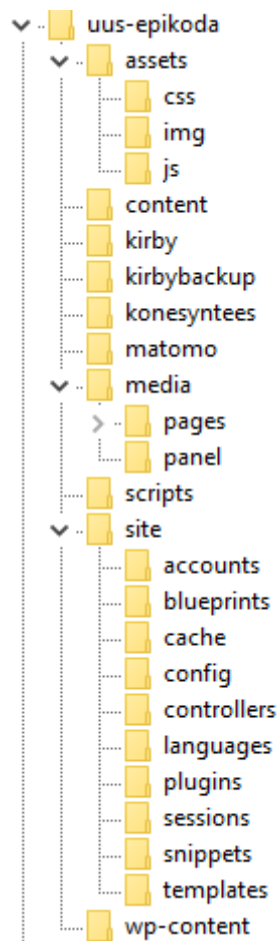
4.5.1 WordPress

EPIKoja eelmine veebileht oli ehitatud kõige levinuma, traditsioonilist tüüpi CMS'i, WordPress'i, peale, mille turuosakaal moodustab teadaolevate sisuhaldustarkvaradega veebilehtedest lausa 60% (W3Techs, 2020). See on üks algajasõbralikumaid võimalusi veebilehe arendusega tegelemiseks, kuna kasutaja ei pea valdama koodi kirjutamise tehnilisi oskuseid ning nii WordPress'i dokumentatsioon kui ka tehniline tugi on väga laialdased ja operatiivsed. Antud viisil veebiarenduse ülemäärane lihtsustamine on andnud tagasilöögi ja teinud keerulisemaks veebilehe tehnilise poole loogika modifitseerimise ja sealsete tõrgete leidmise. Ühe lehekülje funktsionaalsus võib tuleneda arvukatest erinevatest lähtefailidest ning pistikprogrammide, mis igas olukorras korrektselt koos ei toimi. Lisaks eelnevale põhjusele varasema kokkupuute puudumine WordPress'iga sai otsustavaks põhjuseks Kirby CMS valimisel.

4.5.2 Kirby 3

EPIKoja uue veebilehe arenduse jaoks kasutati Kirby 3 CMS'i, mis kujutab endast PHP peale ehitatud failipõhist sisuhaldustarkvara. Kirby peal osutus veebiarendus loogiliseks ja hõlpsasti hallatavaks, kuna selle loodud failide ja kaustade hierarhia on struktureeritud lihtsasti mõistetavalt. Samuti on Kirby'l kogu sisseehitatud funktsionaalsus laialdaselt dokumenteeritud ning probleemide tekkimisel saab nõu küsida nende ametlikus foorumis.

Järgnevalt on toodud kuvatõmmis veebilehe failisüsteemist veebiserveris (vt. Joonis 4.12).

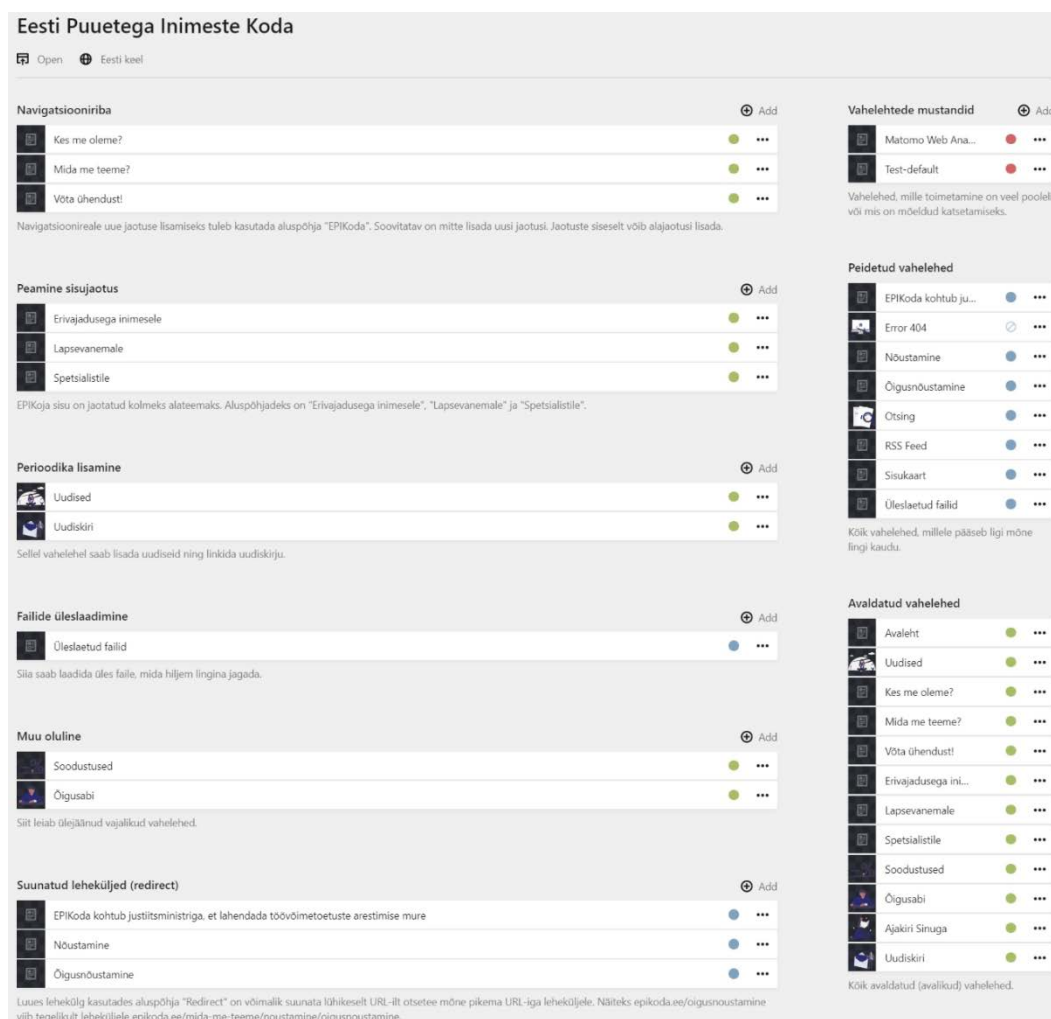


Joonis 4.12: EPIKoja veebilehe failisüsteem

Joonisel kujutatud kataloogidest on tähtsaimad *assets* ja *site* kaustad, mis kujutavad endast peamiseid sisumallide loomise, muutmise ja halduspaneeli kohandamise katalooge. *Assets* kaustas on vastavates alamkaustades talletatud kõik kasutuses olevad CSS ja JavaScript lähtefailid ning veebilehe staatilise osa meediafailid. *Site* kaustas asuvad halduspaneeli registreeritud kasutajate andmed, erinevate funktsionaalsuste kontrollid, pistikprogrammid ning veebilehe kavandid (ingl *blueprint*) ja kujundusmallid (ingl *template*). Lehtede paigutus halduspaneelis ning neile sisestatavate andmete väljad ja tüübid määratakse kavandites (faililaiend *.yml*), see võimaldab kujundusmallis API-stiilis andmetele ligi pääseda. Mainimist väärib ka *snippets* kaust – sinna lähevad koodijupid (ingl *snippet*), mida saab sarnaselt mitmele veebiarendusraamistikule kujundusmallides funktsionaalsete moodulitena kasutada (nt *header* ja *footer*). Kujundusmallid on sisu poolest standardsed HTML failid ühe erisusega – selleks, et neis saaks kasutada Kirby sisseehitatud funktsionaalsust ja erinevaid PHP funktsioone, on nad salvestatud PHP failidena. Loodud lehe sisu kuvamine toimub kavandite, kujundusmallide ja halduspaneelis sisestatud andmete kokku panemisel Kirby poolt.

Välja tasub tuua ka *content*, *media* ja *scripts* kaustad. *Content* kaustas on talletatud kogu halduspaneelis sisestatud info, mida on võimalik antud kaustast ka manuaalselt, ilma halduspaneelita, muuta. *Media* kausta on salvestatud kõik halduspaneelist sisestatud meediafailid (nt pildi- ja videomaterjal, helifailid). *Scripts* kaustas on arendaja poolt kirjutatud spetsiifilisi funktsioone täitvad skriptid (nt tõlkimise tarbeks kogu veebilehe tekstimaterjali eksport).

Kirby pakub võimalust turvakaalutlustel ka halduspaneel välja lülitada, mis tähendab, et veebilehe sisu on muudetav vaid otse serveris *content* kaustas. Kõik Kirby halduspaneeli lehed on kohandatavad (nt kavandites struktuuri muutmine, CSS failis kujunduse muutmine) ja olemuselt väga ligipääsetavad, seega saab ka erivajadusega veebilehe haldaja selle tervet funktsionaalsust kasutada. Joonisel 4.13 on toodud kuvatõmmis Kirby halduspaneelist EPIKoja veebilehel, et paremini demonstreerida selle kohandatavust ja kasutusmugavust.



Joonis 4.13: EPIKoja veebilehe halduspaneel

5 Tulemused

5.1 Teoreetiline lahendus ligipääsetavuse probleemidele

Universaalne disain on suunatud kõigile inimestele, sõltumata nende east, kehakujust või võimetest (Starobogatov, 2017). Üks põhjapanevatest ideedest, mille peale UD on ehitatud, on arusaam, et täiustatud funktsionaalsus tagab kõigile, mitte ainult erivajadustega inimestele, läbi suurenenud ligipääsetavuse ka parema kasutatavuse (Story, 2002).

Universaalse disaini definitsiooni järgi võiks olla iga veebileht, kus on kasutatud UD põhimõtteid, ligipääsetav ja kasutatav kõigi jaoks. Kahjuks on eranditult kõigile sobiva lahenduse pakkumine praktiliselt võimatu. Baker (2014) väidab, et efektiivseima tulemuse saab keskendudes nii kasutajagruppide vahelistele sarnasustele kui ka erinevustele. Kõigepealt tuleks oma tähelepanu koondada universaalsele disainile, et nii indiviidi kui grupi sarnasemad vajadused oleks kaetud ning seejärel spetsiifilisemalt ligipääsetavat disaini rakendada ja veebileht võimalikult paljudele erivajadustele kohandada. See tagab ideaalis kasutajale parema kogemuse kui ainult üldise *one-size-fits-all* lahenduse pakkumine.

Teoreetiliselt on see lihtne ja mõistlik lahendus, kuid päriselus tuleb arvesse võtta ka sellise töö hinda ja keskmise veebiarendaja puudulikke teadmisi ligipääsetava veebi arendamisest. Richards ja Hanson (2004) töid välja, et mõne ettevõtte väitel ei tasu olemasolevate lehekülgede ligipääsetavamaks tegemine ennast ära, kuna ollakse arvamusel, et selline täiendav investeering ei korreleeru otseselt suuremate tuludega. Kuna suurel osal veebiarendajatest jääb oskustest ja kogemustest puudu, et korralikku ligipääsetavat veebimaterjali luua, tundub tõenäoline, et nad ei soovita seda ka oma klientidele. Kuigi leidub kindlasti ettevõtteid, mille äriplaanides on arvestatud erivajadustega inimestega, on ka ettevõtete teadmatus kaudselt probleemiks.

15% maailma rahvastikust on mingisuguse nägemisega seotud erivajadusega, 6% mingisuguse kuulmisega seotud erivajadusega (WHO, 2011), mis tähendab pea kahte miljardit potentsiaalset veebiteenuse kasutajat (Okeke ja Izuogu, 2013). Ligipääsetav veeb tähendab erivajadustega kasutajate jaoks ideaalis paremat kasutajakogemust, mis on otseselt korrelatsioonis kasutajate ülemineku suhtega (ingl *conversion rate*) ning seekaudu ettevõtte tulude kasvuga. Ülemineku suhte analüüs võimaldab kaudselt hinnata ka veebilehe

kasutamise mugavust, mis aitab otsustada kas veebileht vajab täiendavat arendamist või mitte (Garrett, 2010). Seega võib järeldada, et vastupidiselt eelmainitud uuringus välja toodule õigustab veebikeskkonna ligipääsetavaks tegemine tehtud kulutusi.

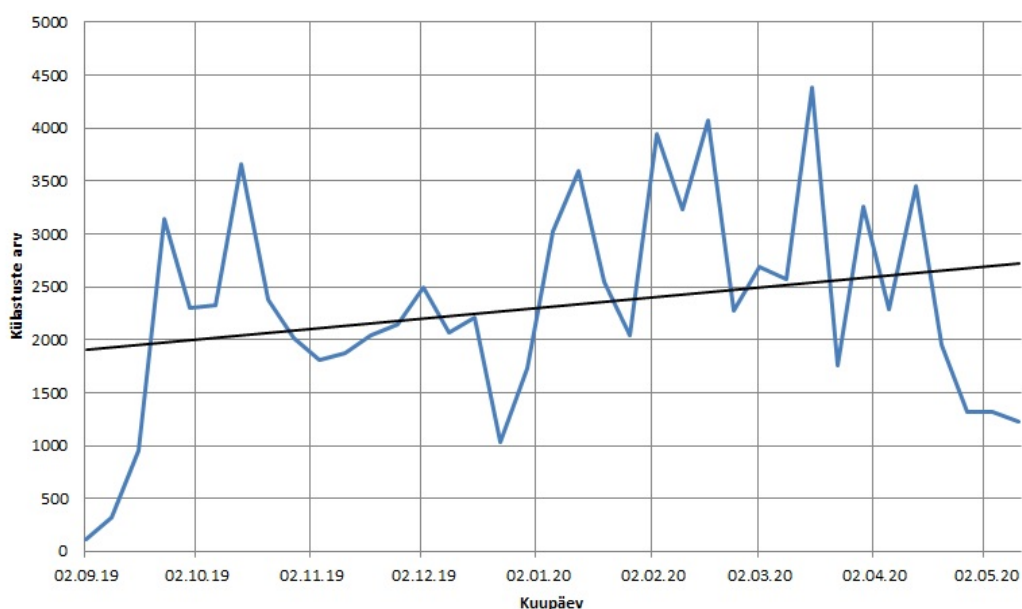
Üks variant olukorra parandamiseks oleks luua universaalne raamistik, mille lisamine mistahes veebilehele võimaldaks seda hõlpsalt ligipääsetavamaks teha. See lubaks ka vähem kogenud veebiarendajatel ilma suure lisapingutusega oma klientidele paremini kasutatavat sisu pakkuda. Kuid arvestades veebiarenduses kasutatavate programmeerimiskeelte ja raamistike hulka ning nende uuenemise kiirust, on sellise lahenduse loomiseks vajaminev aeg ja muud ressursid üüratud. Seega on üpris ebatõenäoline, et midagi sarnast kunagi teostatakse, kuluefektiivsuse tõttu, kellegi teise kui W3C poolt.

Teine võimalus oleks tõsta teadlikkust nii veebiarendajate kui ka nende klientide seas – näiteks korraldada rohkem koolitusi ligipääsetavuse teemal ja rõhutada selle olulisust. Kuigi olemasolevate veebilehtede ligipääsetavuse tõstmine võib osutuda keeruliseks, teeb sellega algusest peale arvestamine töö tehniliselt kergemaks ja kogu arendusprotsessi kõrval ei tohiks täiendav ajakulu olla suur. Antud juhul ei tõstaks ligipääsetavus ka palju veebiarenduse projekti maksumust ning ettevõtetal oleks rohkem motivatsiooni selle täide viimiseks. Selline lahendus oleks pikas perspektiivis tõenäoliselt tõhusam ja ka üldiselt kasulik, kuna kliendid oskaksid soovitud funktsionaalsust arendajale paremini väljendada.

5.2 EPIKoja kasutajate analüütika

Matomo veebiliikluse analüütika pistikprogramm sai paigaldatud arenduse käigus, et oleks võimalik andmeid koguda alates veebilehe avalikustamisest. Matomo Web Analytics on ainus pistikprogramm kogu veebilehel, mis töötamise käigus andmebaasi kasutab.

Joonisel 5.1 on toodud tabel igakuisest külastajate arvust (sinine) veebilehe avamisest 2019. aasta septembris kuni 2020. aasta maini. Matomo andmete põhjal on selles ajavahemikus keskmiselt nädalas 2432 unikaalset külastust. Graafikule on kuvatud ka trendijoon (must), millest võib järeldada, et veebilehe populaarsus on ajas järk-järgult kasvamas.

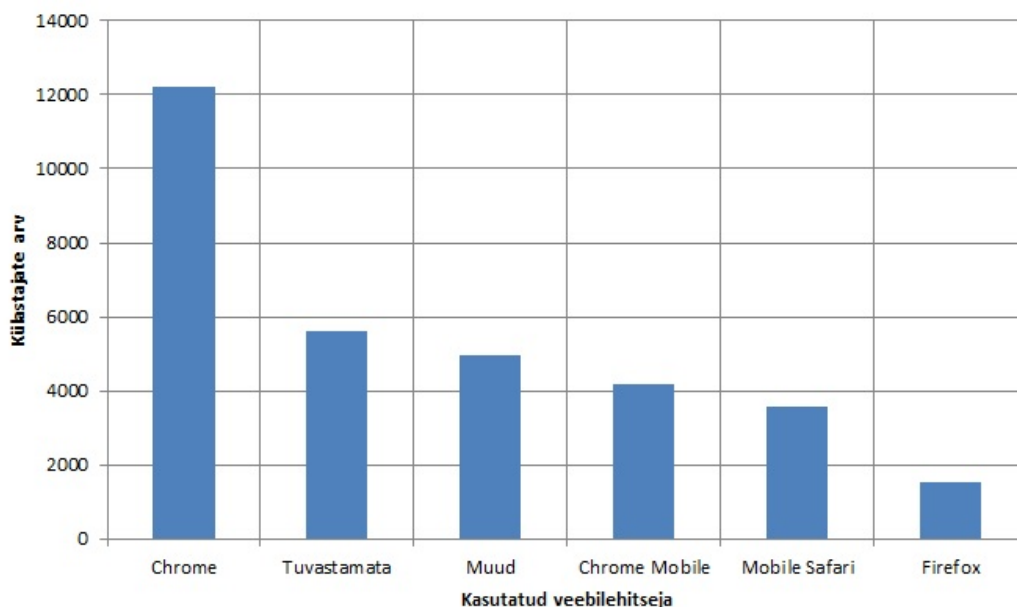


Joonis 5.1: EPIKoja külastajate arv 2019-2020

Keskmine külastaja veedab EPIKoja veebilehel 2 minutit ja 5 sekundit, mille jooksul teeb külastaja keskmiselt 2.9 toimingut (nt laeb faili alla, liigub järgmisele alalehele, klikib lingile).

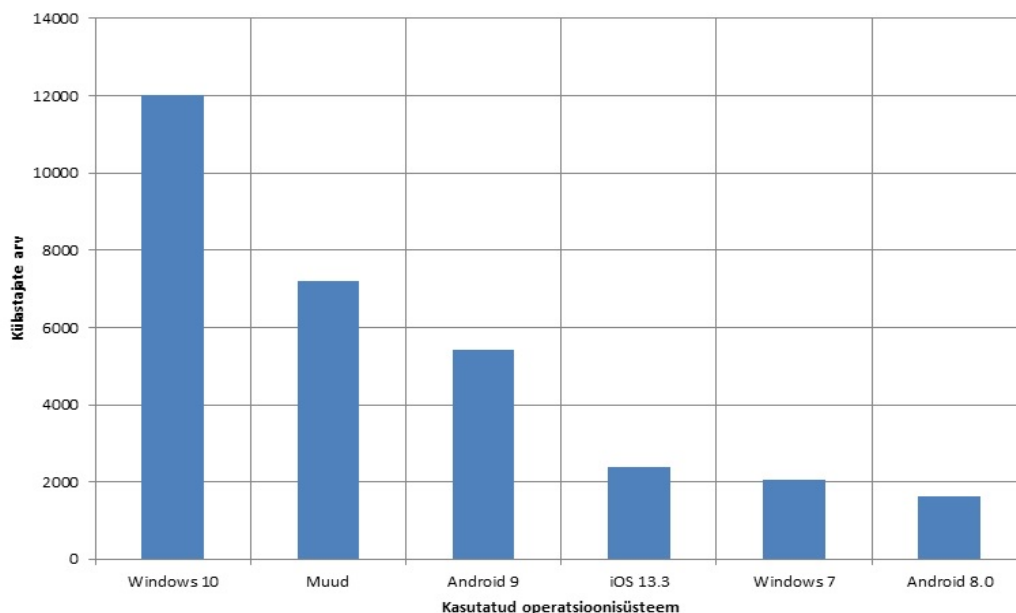
Peamised seadmed, mida EPIKoja külastajad kasutavad, on lauaarvutid ja sülearvutid, mis moodustavad kõigist külastajate seadmetest vastavalt 52.4% ja 42.1%. Neile järgnev seadme tüüp on tahvelarvuti, mille osakaal moodustab 5.1%. Ülejäänud seadmete tüüpi ei suutnud Matomo moodul tuvastada.

Joonisel 5.2 on toodud statistika EPIKoja veebilehe külastamiseks kasutatud veebilehitsejatest. Levinuim neist on Google Chrome, moodustades kokku 38% kõikidest kasutatud veebilehitsejatest. Sellele järgnevad Chrome Mobile, Mobile Safari ja Firefox, vastavalt 13%, 11% ja 5% osakaaludega. Suure osa moodustavad ka veebilehitsejad, mida Matomo moodul ei suutnud tuvastada.



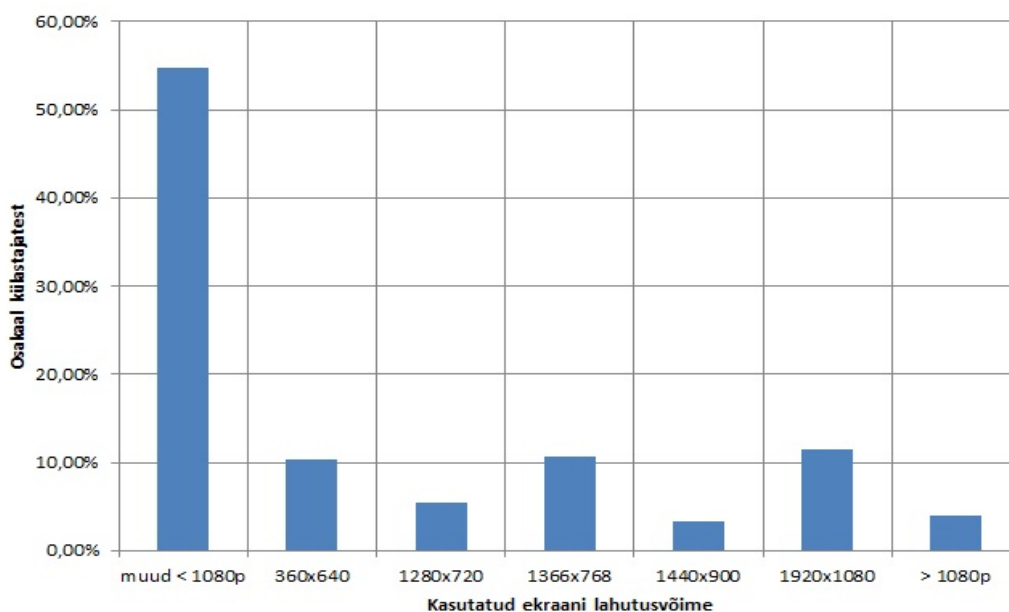
Joonis 5.2: EPIKoja külastamisel enim kasutatud veebilehitsejad

Joonisel 5.3 on toodud statistika EPIKoja veebilehe külastamiseks kasutatavate seadmete operatsioonisüsteemidest. Kasutuses on peamiselt Windows 10 ja Android 9 vastavalt 37% ja 17% osakaaludega. Märkimist väärivad ka Windows 7, Android 8.0 ja iOS 13.3, mis moodustavad kokku 18% kõikidest kasutatud operatsioonisüsteemidest.



Joonis 5.3: EPIKoja külastamisel enim kasutatud operatsioonisüsteemid

Ekraanide eraldusvõimet saab kirjeldada pikslites (px), kus esitatakse lahutusvõime kujul *laius(px) x kõrgus(px)*. Erineva eraldusvõimega ekraane on EPIKoja külastajate seas kirjutamise hetkeks olnud 464, kusjuures suure osa neist moodustavad ebaharilikud eraldusvõimed (nt 1224x685, 375x667). Antud põhjusel on külastajate ekraanide eraldusvõimed lisaks tähtsamatele grupeeritud kahte jaotusesse: väiksem kui 1080 px ekraanilaius ja suurem kui 1080 px ekraanilaius (vt. Joonis 5.4). Grupeeritud andmetest on välja jäetud tähtsaimad eraldusvõimed, st “muud < 1080p” grupp ei sisalda endas eraldi välja toodud “1366x768”, “360x640”, “1280x720” ja “1440x900” eraldusvõimeid, et vältida andmete duplitseerimist.



Joonis 5.4: EPIKoja külastamisel enim kasutatud eraldusvõimed

Veebiliikluse analüütika on väga võimekas tööriist, mis ei paku enam kõigest paari näitajat ettevõtte veebilehe edukuse hindamiseks, vaid kogub ka ohtralt erinevaid andmeid kasutajate liikumise ja toimingute kohta, mis võimaldab paremini mõista veebilehe külastajate käitumist. Arvukate näitajate ja hinnangute analüüsimisel tuleb kindlasti arvestada kliendi vajadustega, vältides analüüsiaruandes informatsiooni üleküllust. Külastajate veebilehitsejate osakaalude analüüs võimaldab ettevõttel hinnata, kas veebilehe kujundus vajab kohandamist populaarsemate veebilehitsejate jaoks, mis tuleneb sellest, et kujundusvõtted võivad anda erinevates veebilehitsejates isemoodi tulemusi. Andmed kasutaja veebilehele jõudmise kohta, st mis otsingusõnaga lehele jõuti või kust veebilehelt kasutaja sinna suunati, on samuti olulised. Teades mida kasutajad ettevõtte veebilehelt peamiselt otsivad ning mis teed pidi sinna jõutakse, saab ettevõtte veebilehte paremini otsingumootoritele optimiseerida (ingl SEO – *Search Engine Optimization*) ning luua kvaliteetsemaid partnerlussuhteid kasutajaid neile suunanud ettevõtetega (Phippen, Sheppard ja Furnell, 2004).

5.3 Ligipääsetavuse, jõudluse ja turvalisuse hindamine

5.3.1 Automatiseeritud testid

Kasutatud testimistarkvaradele on viidatud töö lõpus kasutatud allikate nimekirjas.

Loodud veebilehe turvalisuse testimiseks kasutati järgnevaid vabavaralisi tarkvarasid: Web Cookies Scanner, SSLabs Qualys SSL Server Test, ImmuniWeb Website Security Test ja Pentest-Tools Website Scanner. Esimesed kolm meetodit andsid väljastatud aruandes tähelise hinnangu, vastavalt A, B ja A+. Pentest-Tools Website Scanner liigitas EPIKoja veebilehe madala riskiga gruppi, kus läbiti sooritatud kümnest testist edukalt kaheksa.

Ligipääsetavuse testimiseks kasutati kahte vabavaralist tarkvara: WebAIM WAVE ja veebilehitseja pistikprogramm axe: The Accessibility Engine. Mõlemad meetodid tõid välja mõned korduvad vähemtähtsad vead, mille parandamisega tulevikus tegeleda.

Jõudlustestideks kasutati kolme vabavaralist tarkvara: Pingdom Website Speed Test, GTMetrix Website Speed ja Dareboost Website Speed and Analysis Test. Antud meetodid andsid 100-punktilisel skaalal hinnanguteks vastavalt 84, 88 ja 80 punkti. Asjaolud, mis takistasid kõrgema punktisumma saamist, tulenesid peamiselt teekide ja kirjatüüpide importimisest välistest allikatest majutusserveri asemel.

Veebilehel on erinevate testide aruannete põhjal tehtud arvukalt parandusi ja optimeerimist, peamised neist lehe sisu veebilehitseja vahemällu salvestamine (ingl *browser caching*) ning lehe sisu serveris enne veebilehitsejale saatmist tihendamine (ingl *gzip compression*). Veebilehel olev pildimaterjal optimeerimist ei vajanud, kuna on kasutatud SVG (skaleeritav vektorgraafika, ingl *Scalable Vector Graphics*) faile, mis on kodeeritud kadudeta formaadis.

Kokkuvõttes võib automaatsete tulemustega rahule jääda, loodud veebileht on enamikust turvalisem, parema jõudlusega ning väga ligipääsetav.

5.3.2 Kasutajate tagasiside

Esimene samm ligipääsetavuse hindamiseks on veebilehe analüüsimine ilmselgete puudujääkide leidmiseks. Sealt edasi liigutakse automaatsete testimissüsteemide juurde, mis töötavad peamiselt erinevate ligipääsetavuse juhendite põhjal ning keskenduvad antud juhendites välja toodud nõuetele vastamisele. Viimane ja tähtsaim osa on kasutajate kaasamine protsessi, mida peetakse laialdaselt veebi ligipääsetavuse hindamise kuldstandardiks (Vigo et al., 2014). Kasutajate vahetu tagasiside võimaldab leida ja juhtida arendajate tähelepanu probleemidele, millele automaatsed testimissüsteemid ei ole tähelepanu pööranud.

Samal põhjusel liiguti peale EPIKoja veebilehel manuaalsete ja automaatsete testide läbi viimist kasutajate tagasiside poole. Koostöös EPIKoja peaspetsialistiga loodi veebilehe kasutusmugavusele ja funktsionaalsusele tagasiside saamiseks UT LimeSurvey (Schmitz, 2020) keskkonnas küsitlus. Küsimustik koostati kolmes keeles (eesti, inglise ja vene) ning hõlmab endas 17 ligipääsetavusega seotud küsimust. Küsimustest üheksa on valikvastustega, kuus vaba teksti vormis, üks punktiskaalas hinnangu andmine ja üks numbrilise sisendiga küsimus. Küsimustik aktiveeriti ja tehti kasutajatele kättesaadavaks käesoleva aasta märtsis ning on siamaani üleval. Kokku on kirjutamise hetkeks ära saadetud 77 vastust, millest 3 olid erinevatel aegadel küsimustiku korrektse töötamise kontrollimiseks tehtud testid ning 74

Google'i otsingu indekseerimise robotite poolt tekitatud valehäired (s.t. ebatäielikud vastused). Sellel põhjusel käesoleval hetkel adekvaatset tagasisidet, mida analüüsida ja kasutada EPIKoja veebilehe parendamiseks, kasutada ei ole. Tagasisideküsitlus on lisatud kolmes keeles käesoleva bakalaureusetöö kausta.

6 Kokkuvõte

Käesoleva bakalaureusetöö praktilise osa eesmärgiks oli Eesti Puuetega Inimeste Koja veebileht kaasajastada ning viia ligipääsetavuse standarditega vastavusse. Tööprotsess hõlmas endas eelneva veebilehe sisu üle toomist, arenduseks vajalike tarkvarade ja meetodite leidmist ning seejärel nende kasutamist. Veebilehe valmimisel hinnati seda ulatuslikult ligipääsetavuse, jõudluse ja turvalisuse vaatenurkadest automatiseeritud testidega. Loodi ka veebilehe kasutusmugavuse kohta tagasiside saamise eesmärgil küsitlus, mis kahjuks EPIKoja külastajate hulgas huvi ei äratanud.

Töö teoreetiline osa kujutas endast veebi ligipääsetavuse valdkonna tutvustamist, seal aktuaalsete probleemide analüüsimist ning potentsiaalse lahenduse pakkumist. Täpsemalt arutleti ligipääsetavuse üle erivajadustega veebikasutaja vaatenurgast ning seletati, mida arendaja teha saab, et luua võimalikult ligipääsetav veebikeskkond. Viimases osas toodi välja erinevaid näitajaid EPIKoja veebilehe külastajate kohta.

Autori edasised plaanid seoses EPIKoja veebilehe arendusega hõlmavad endas järk-järgult automaatsete aruannete põhjal puudujääkide kõrvaldamist ning veebilehele funktsionaalsuste lisamist. Üks suurematest lisafunktsionaalsustest, mida plaanitakse lähiajal EPIKoja veebilehele lisada, on mitmekeelsus (ingl *multilingual support*). Selle tarbeks kirjutati Kirby dokumentatsiooni põhjal eraldiseisev PHP skript, mis väljastab terve veebilehe tekstilise sisu Javascript'i objektide notatsioonis (ingl JSON – *JavaScript Object Notation*). Antud formaadis on EPIKojaga koostööd tegeval tõlkebürool võimalik hõlpsalt tõlkeprogrammis veebilehe sisu tõlkida.

Kasutatud allikad

Baker, S. C., (2014), “*Making It Work for Everyone: HTML5 and CSS Level 3 for Responsive, Accessible Design on Your Library's Web Site*“, Journal of Library & Information Services in Distance Learning, 8:3-4, 118-136 [Võrgumaterjal] [Kasutatud 09.11.2019] <https://www.tandfonline.com/doi/full/10.1080/1533290X.2014.945825>

Bastian Allgeier GmbH, (2020), Kirby sisuhaldustarkvara (versioon 3), [Võrgumaterjal] Kättesaadav: <https://getkirby.com/>

Caldwell et al., (2008), “*Web Content Accessibility Guidelines (WCAG) 2.0*“, World Wide Web Consortium (W3C), [Võrgumaterjal] [Kasutatud: 12.05.2020] <https://www.w3.org/TR/WCAG20/>

Chisholm, W., Vanderheiden, G., Jacobs, I., (1999), “*Web Content Accessibility Guidelines 1.0*“, World Wide Web Consortium (W3C), [Võrgumaterjal] [Kasutatud: 12.05.2020] <https://www.w3.org/TR/WAI-WEBCONTENT/>

Clark, J., (2006), “*To Hell with WCAG 2*“, A List Apart 217, [Võrgumaterjal] [Kasutatud 16.05.2020], <https://alistapart.com/article/tohellwithwcag2/>

Common European Research Classification Scheme (CERCS) Teadusvaldkondade ja -erialade klassifikaator [Võrgumaterjal] [Kasutatud 26.10.2019] <https://www.etis.ee/Portal/Classifiers/Details/d3717f7b-bec8-4cd9-8ea4-c89cd56ca46e>

Cooper et al., (2018), “*Web Content Accessibility Guidelines (WCAG) 2.1*“, World Wide Web Consortium, [Võrgumaterjal] [Kasutatud 11.05.2020] <https://www.w3.org/TR/WCAG21>

Dareboost, (2020), Website Speed and Analysis Test, Jõudluse testimise tarkvara, [Võrgumaterjal] [Kasutatud 13.05.2020] Kättesaadav: <https://www.dareboost.com/en>

Deque Systems Inc., (2020), axe: The Accessibility Engine, Ligipääsetavuse testimise tarkvara, [Võrgumaterjal] [Kasutatud 13.05.2020] Kättesaadav: <https://www.deque.com/axe/>

Faulkner, S., O'Hara, S., (2020), “*ARIA in HTML*“, World Wide Web Consortium Working Draft, [Võrgumaterjal] [Kasutatud 07.05.2020] <https://www.w3.org/TR/html-aria/>

Free Software Foundation, Inc., (2007), Matomo Web Analytics, veebiliikluse analüütika pistikprogramm, [Võrgumaterjal] Kättesaadav: <https://matomo.org/>

Garrett, J. J., (2010), *“The Elements of User Experience: User-Centered Design for the Web and Beyond, Edition 2”*, Pearson Education

GTMetrix, (2020), Website Speed, Jõudluse testimise tarkvara, [Võrgumaterjal] [Kasutatud 13.05.2020] Kättesaadav: <https://gtmetrix.com/>

Hartl et al., (2018), Javascript Cookie (js-cookie), Javascript'i teek küpsiste loomiseks, [Võrgumaterjal] Kättesaadav: <https://github.com/js-cookie/js-cookie>

High-Tech Bridge SA, (2020), Website Security Test, Turvalisuse testimise tarkvara, [Võrgumaterjal] [Kasutatud 13.05.2020] Kättesaadav: <https://www.immuniweb.com/websec/>

Kasik, R. (2007) *“Sissejuhatus tekstiõpetusse”*, Tartu Ülikooli kirjastus.

Kinsta Inc., (2020), *“What is a Content Management System (CMS)?”*, [Võrgumaterjal] [Kasutatud 11.05.2020], <https://kinsta.com/knowledgebase/content-management-system/>

Kirkpatrick et al, (2018), *“Web Content Accessibility Guidelines (WCAG) 2.1 Editor's Draft”*, [Võrgumaterjal] [Kasutatud 13.05.2020] <https://w3c.github.io/wcag21/understanding/21/text-spacing.html>

Koolmeister, H., (2016), *“Ligipääsetavusest digitaalkeskkonnas”*, Sotsiaalministeerium. [Võrgumaterjal] [Kasutatud 26.10.2019] https://www.mkm.ee/sites/default/files/dhn_2016_04_13_ligipaasetavus.pdf

Lazar, J., Dudley-Sponaugle, A., Greenidge, K-D., (2004), *“Improving web accessibility: a study of webmaster perceptions”*, Computers in Human Behavior, 20(2), 269-288, [Võrgumaterjal] [Kasutatud 27.10.2019] <https://www.sciencedirect.com/science/article/pii/S0747563203000906>

Longman, A. W., (1998), *“A history of HTML”*, World Wide Web Consortium, [Võrgumaterjal] [Kasutatud 28.10.2019], <https://www.w3.org/People/Raggett/book4/ch02.html>

Maldonado, L., (2018), *“What's the Document Object Model, and why you should know how to use it.”*, [Võrgumaterjal] [Kasutatud 11.05.2020] <https://www.freecodecamp.org/news/whats-the-document-object-model-and-why-you-should-know-how-to-use-it-1a2d0bc5429d/>

Nested Accordion, (2014), Javascript'i teek teksti liigendamiseks, [Võrgumaterjal]
Kättesaadav: <http://www.adipalaz.com/>

Okeke, O., Izuogu, D., (2013), "*Issues affecting implementation of Web Accessibility Guidelines*", Sheffield Hallam University, [Võrgumaterjal] [Kasutatud 28.10.2019]
<https://pdfs.semanticscholar.org/5397/e016cecee87b4b8b71658f8c7570d8ae8c46.pdf>

Pang et al., (2016), "*Directing user attention via visual flow on web designs*", ACM Transactions on Graphics, 35(6), [Võrgumaterjal] [Kasutatud 08.05.2020]
<https://dl.acm.org/doi/pdf/10.1145/2980179.2982422>

Pentest-Tools, (2020), Website Vulnerability Scanner, Turvalisuse testimise tarkvara, [Võrgumaterjal] [Kasutatud 13.05.2020] Kättesaadav: <https://pentest-tools.com/website-vulnerability-scanning/website-scanner>

Phippen, A., Sheppard, L., Furnell, S., (2004), "*A practical evaluation of Web analytics*", [Võrgumaterjal] [Kasutatud 14.05.2020]
<https://www.emerald.com/insight/content/doi/10.1108/10662240410555306/full/html#idm45867249785888>

Power et al., (2012), "*Guidelines are Only Half of the Story: Accessibility Problems Encountered by Blind Users on the Web*", SIGCHI Conference on Human Factors in Computing Systems (CHI 2012), 433-442 [Võrgumaterjal] [Kasutatud 10.05.2020]
<https://dl.acm.org/doi/10.1145/2207676.2207736>

Qualys Inc., (2020), SSL Server Test, Turvalisuse testimise tarkvara, [Võrgumaterjal] [Kasutatud 13.05.2020] Kättesaadav: <https://www.ssllabs.com/ssltest>

Refsnes Data, (2020), "*HTML Semantic Elements*", W3Schools, [Võrgumaterjal] [Kasutatud 17.05.2020], https://www.w3schools.com/html/html5_semantic_elements.asp

Richards, J. T., Hanson, V. L., (2004), "*Web Accessibility: A Broader View*", 72-79, WWW '04 Proceedings of the 13th international conference on the World Wide Web [Võrgumaterjal] [Kasutatud 01.12.2019], <https://dl.acm.org/citation.cfm?id=988683>

Rogojan, B., (2019), "*Introduction to Flat-File CMS: Deciding Whether They Are Right for You*", ButterCMS, [Võrgumaterjal] [Kasutatud 11.05.2020]

<https://buttercms.com/blog/introduction-to-flat-file-cms-deciding-whether-they-are-right-for-you>

Rotondi et al., (2013), “*Critical Design Elements of E-Health Applications for Users With Severe Mental Illness: Singular Focus, Simple Architecture, Prominent Contents, Explicit Navigation, and Inclusive Hyperlinks*“, Schizophrenia Bulletin, 41(2), 440-448, [Võrgumaterjal] [Kasutatud 08.05.2020]

<https://academic.oup.com/schizophreniabulletin/article/41/2/440/2526083>

Schmitz, C., (2020), Tartu Ülikooli LimeSurvey veebipõhiste küsimustike koostamise keskkond, [Võrgumaterjal] [Kasutatud 20.02.2020] Kättesaadav: <https://survey.ut.ee/>

Sierkowski, B., (2002), “*Achieving web accessibility*“, Proceedings of the 30th annual ACM SIGUCCS conference on User services (SIGUCCS '02), 288-291, [Võrgumaterjal] [Kasutatud 28.10.2019] <http://doi.acm.org/10.1145/588646.588725>

Sloan et al., (2016), “*Accessibility 2.0: People, Policies and Processes*“, [Võrgumaterjal] [Kasutatud 14.05.2020] <https://dl.acm.org/doi/pdf/10.1145/1243441.1243471>

SolarWinds WorldWide LLC, (2020), Pingdom Website Speed Test, Jõudluse testimise tarkvara, [Võrgumaterjal] [Kasutatud 13.05.2020] Kättesaadav: <https://tools.pingdom.com/>

Starobogatov, E., (2017), “*Universaalse disaini hindamisvahendi disain ja haldamine*“, Tallinna Ülikool [Võrgumaterjal] [Kasutatud 28.10.2019]

statcounter GlobalStats, (2019), “*Desktop Screen Resolution Stats Worldwide*“, [Võrgumaterjal] [Kasutatud 09.05.2020] <https://gs.statcounter.com/screen-resolution-stats/desktop/worldwide/#yearly-2019-2019-bar>

Story, M. F., (2002), “*Distance Education in Universal Design*“, Universal Design: 17 Ways of Thinking and Teaching, [Võrgumaterjal] [Kasutatud 28.10.2019], <http://biblioteket.husbanken.no/arkiv/dok/853/rapport2.pdf>

Swallow, D. M., (2017), “*Understanding and Supporting Web Developers: Working Practices and Resources for the Creation and Evaluation of Accessible Websites*“, University of York, [Võrgumaterjal] [Kasutatud 28.10.2019], <https://pdfs.semanticscholar.org/63e9/41d0ab64a9e4fb61196d044197384e47a841.pdf>

Swiper, (2020), Javascript'i teek karusselli loomiseks, [Võrgumaterjal] Kättesaadav: <https://swiperjs.com/>

Zuzana, P., (2016), “Why Faster Page Load Time is Better For Your Website [Infographic]“, [Võrgumaterjal] [Kasutatud 08.05.2020] <https://skilled.co/resources/speed-affects-website-infographic/>

The jQuery Foundation, (2020), jQuery, jQuery Validate, Javascripti teegid üldise lisafunktsionaalsusega, [Võrgumaterjal] Kättesaadav: <https://jquery.com/>

University of Connecticut, (2018), “What is a screen reader and how does it work?“, [Võrgumaterjal] [Kasutatud 08.05.2020] <https://accessibility.its.uconn.edu/2018/08/22/what-is-a-screen-reader-and-how-does-it-work/>

W3Techs, (2020), “Usage statistics and market share of WordPress“, [Võrgumaterjal] [Kasutatud 11.05.2020] <https://w3techs.com/technologies/details/cm-wordpress>

Vandenbark, R. T., (2010), “Tending a Wild Garden: Library Web Design for Persons with Disabilities“, Information Technology & Libraries, 29(1), 23–29, [Võrgumaterjal] [Kasutatud 26.10.2019] <https://pdfs.semanticscholar.org/9faa/b338e395283c19ef3f6b214a1c6129eac803.pdf>

WebAIM, (2020), Contrast Checker, Värvikontrasti analüsaator, [Võrgumaterjal] [Kasutatud 11.05.2020] Kättesaadav: <https://webaim.org/resources/contrastchecker>

WebAIM, (2020), WAVE Website Accessibility Evaluation Tool, Ligipääsetavuse testimise tarkvara, [Võrgumaterjal] [Kasutatud 13.05.2020] Kättesaadav: <https://wave.webaim.org/>

WebCookies Scanner, (2020), Turvalisuse testimise tarkvara, [Võrgumaterjal] [Kasutatud 13.05.2020] Kättesaadav: <https://webcookies.org>

Vigo et al., (2014), “Are users the gold standard for accessibility evaluation?“, Proceedings of the 11th Web for All Conference, 1-4, [Võrgumaterjal] [Kasutatud 10.05.2020] <https://dl.acm.org/doi/abs/10.1145/2596695.2596705>

WordPress Foundation, (2020), WordPress sisuhaldustarkvara, [Võrgumaterjal] Kättesaadav: <https://et.wordpress.org/download/>

World Health Organization, (2011), “*World report on disability*“, Genf [Võrgumaterjal]
[Kasutatud 01.12.2019], https://www.who.int/disabilities/world_report/2011/factsheet.pdf

Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, **Mattias Tõnisson**

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

“Veebilehe ligipääsetavus Eesti Puuetega Inimeste Koja veebilehe arendusprotsessi näitel”,

mille juhendaja on **Helle Hein**,

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Mattias Tõnisson
20.05.2020